



**Universidade de Brasília
Instituto de Artes
Departamento de Design**

**ENCORE:
Projeto de jogo de Plataforma e Ritmo**

Pedro Braga Vasconcelos

Brasília – DF
2017

Pedro Braga Vasconcelos

ENCORE:

Projeto de jogo de Plataforma e Ritmo

Relatório apresentado ao Departamento de Design da Universidade de Brasília – UnB como requisito parcial para aprovação na Diplomação em Programação Visual, orientado pelo professor Tiago Barros Pontes e Silva

Brasília – DF

2017

RESUMO

O relatório a seguir descreve o passo a passo para o desenvolvimento de um jogo digital com foco na experiência final de usuário, desde sua concepção inicial até o resultado final. O projeto pertence ao gênero de Plataforma e Ritmo e tem a intenção de se diferenciar de similares, expandindo as possibilidades para o gênero. O método demanda uma visão definida de como o projeto final deve ser. A partir dessa visão, desse objetivo, são discutidos os conceitos e pesquisas envolvidos na produção do jogo, e os elementos individuais que o compõem, todos voltados para a melhor realização da intenção inicial.

Palavras-chave: Jogo, videogame, ritmo, *high-concept*, experiência.

LISTA DE IMAGENS

Figura 01 - Tennis for Two, o primeiro jogo digital.....	1
Figura 02 - High-Concept do projeto Terracotta.....	6
Figura 03 - Personagens existentes no produto final, demonstrando o estilo visual final.....	6
Figura 04 - High-Concept do jogo Umbra.....	7
Figura 05 - Resultado final do projeto.....	7
Figura 06 - O high-concept do projeto.....	9
Figura 07 - O jogo Super Mario World (1990), da Nintendo.....	11
Figura 08 - O jogo Sonic the Hedgehog (1991), desenvolvido pela SEGA.....	12
Figura 09 - O jogo Hatsune Miku Project Diva: Future Tone (2016), da SEGA.....	13
Figura 10 - O jogo Super Mario Bros. (1985), da Nintendo.....	15
Figura 11 - The Impossible Game.....	16
Figura 12 - Geometry Dash.....	17
Figura 13 - Bit.Trip Runner.....	18
Figuras 14 e 15 - Plataformas que alternam entre si de acordo com o ritmo da música.....	19
Figura 16 - Uma cena do jogo Jak and Daxter.....	21
Figura 17 - Os protagonistas da franquia Kingdom Hearts, Sora, Pato Donald e Pateta.....	22
Figura 18 - O clássico salão de dança do filme A Bela e a Fera (1991), representado no jogo Kingdom Hearts II (2005)	23
Figura 19 - Um dos cenários do jogo Yooka-Laylee.....	24
Figura 20 - Alguns painéis do programa Photoshop CS6.....	25
Figura 21 - O Animate.....	26
Figura 22 - Um desenho sendo realizado no Krita.....	27
Figura 22 - Ciclo de caminhada, animação tradicional.....	32
Figura 23 - Super Mario World (1990), Nintendo.....	33
Figura 24 - Sonic Mania (2017), da SEGA.....	33
Figura 25 - Inbetweening no Adobe After Effects.....	34
Figura 26 - O jogo Broken Age (2015), de Tim Schafer.....	35
Figura 27 - Concept Art de uma criatura chamada Hipogrifo, para o filme Harry Potter e o Prisioneiro de Azkaban (2003), dos estúdios Warner Bros.....	36

Figura 28 - Um tipo de Concept-Art para o filme Frozen (2013), de Walt Disney, destrinchando possíveis expressões faciais de uma das protagonistas.....	37
Figura 29 - A Sprite-sheet do protagonista do jogo Megaman (1987), Capcom.....	39
Figura 30 - Uma Tileset básica de um cenário	40
Figura 31 - A marca do jogo.....	43
Figura 32 - Versões 1 e 2 do protagonista. A versão 2 foi utilizada para o <i>High-Concept</i>	45
Figura 33 - O High-concept do jogo, outra vez.....	46
Figura 34 - Teste de estilo para o protagonista.....	47
Figura 35 - Versão 3, parte do teste feito no Krita. A imagem inteira será discutida mais adiante.....	48
Figura 36 Teste final do personagem.....	49
Figura 37 - Primeiro teste para o rosto de Woogie.....	50
Figura 38 - Versão final de Woogie.....	51
Figura 39 - Primeiro esboço de cenário.....	52
Figura 40 - Teste realizado no Krita, em sua forma completa.....	53
Figura 41 - Símbolo de uma Colcheia.....	53
Figura 42 - Esboço dos inimigos, ao lado do personagem para comparação de tamanho.....	54
Figura 43 - Boogie adolescente.....	55
Figura 44 - Os 3 passos para a criação da animação.....	56
Figura 45 - Cada ação que o personagem pode executar. Da esquerda para a direita: Correr, pular, cair, ficar parado e sofrer danos.....	57
Figuras 46 e 47 - Spritesheet do protagonista, com a representação de cada célula abaixo, em tons de roxo.....	58
Figura 48 - Peças do cenário.....	59
Figuras 49 e 50 - As peças da cachoeira.....	60
Figura 51 - Teste realizado no Photoshop, dimensões de 1920 x 1080 pixels.....	62
Figura 52 - Versões finais dos robôs.....	63
Figura 53 - Teste 1 - 100% do tamanho.....	64
Figura 54 - Teste 2, à 50% do tamanho.....	65
Figura 55 - Teste 3, à 1/3 do tamanho.....	65
Figura 56 - Representação da primeira fórmula no código do jogo.....	66

Figura 57 - Representação da segunda fórmula no código do jogo, especificamente no objeto do alto-falante maior, no cenário.....	66
Figura 58 - O código do contador.....	67
Figura 59 - O valor do contador sendo aplicado em uma plataforma móvel. O contador controla a variável que determina a direção do movimento da plataforma.....	67
Figura 60 - O sol representando a nota do jogador.....	68
Figura 61 - Tela inicial.....	69
Figura 62 – Os personagens, juntos.....	70
Figura 63 – Cena do jogo.....	70
Figura 64 – A contagem de pontos ao final de cada fase.....	71

SUMÁRIO

1. INTRODUÇÃO.....	1
2. METODOLOGIA.....	3
3. PROPOSTA DE JOGO E HIGH-CONCEPT.....	5
3.1. O que é High-Concept.....	5
3.2. A Proposta.....	8
4. CONCEITUAÇÃO.....	9
4.1. Pesquisa e Planejamento.....	9
4.1.1. O que é Jogo.....	9
4.1.2. Gênero de jogo.....	10
4.1.3. O que é Mecânica.....	13
4.1.4. Referências de Mecânica.....	15
4.1.5. Referências Visuais.....	19
4.1.6. Referências Musicais.....	24
4.1.7. Pesquisa de Ferramentas.....	25
4.1.8. Tipos de Animação.....	30
4.2. Elementos de Pré-produção.....	35
4.2.1. Concept-art.....	35
4.2.2. Enredo.....	37
4.3. Elementos de Produção.....	38
4.3.1. Assets.....	38
4.3.2. Programação, Implementação, Erros e Testes.....	40
5. DESENVOLVIMENTO.....	41
5.1. Pré-produção.....	41
5.1.1. História.....	41
5.1.2. Personagens.....	44
5.1.3. Cenário.....	51
5.1.4. Miscelânea.....	53
5.2. Produção.....	55
5.2.1. Assets.....	55
5.2.1.1. Boogie.....	55
5.2.1.2. Cenário.....	58
5.2.1.3. Música.....	61

5.2.1.4. Inimigos.....	62
5.2.2. Programação.....	63
5.2.2.1. Fase de Testes.....	63
5.2.2.2. Câmera, Zoom, Teste de Tamanho.....	64
5.2.2.3. Ritmo.....	66
5.2.2.4. Morte e Pontos.....	68
6. RESULTADO.....	70
6.1. Tela Inicial.....	70
6.2. O jogo.....	70
6.3. Tela de Resultado.....	71
7. CONCLUSÃO.....	72
8. REFERÊNCIAS.....	73

1. INTRODUÇÃO

Videogames são uma mídia caracterizada por ser interativa, uma forma de entretenimento digital. Embora mais recentes que Livros, Jornais e filmes, o primeiro exemplo de um Videogame existiu desde os anos cinquenta, sendo ele o jogo *Tennis for Two*, criado por William Higinbotham em um osciloscópio.

Quase sessenta anos depois da criação de *Tennis for Two*, videogames se tornaram um dos mercados mais lucrativos atuais, valendo, apenas nos Estados Unidos da América, 18 bilhões de dólares. Como mídia, os jogos evoluíram bastante, se tornando mais complexos de se fazer e de se jogar, e providenciando experiências diversas e únicas. Antes um simples aparelho que simula um jogo de tênis, jogos agora podem oferecer simulações complexas de mundos e histórias que se desenvolvem por horas. Eles também oferecem sons e imagens que chegam próximos à vida real, ou não, dependendo da intenção dos criadores.

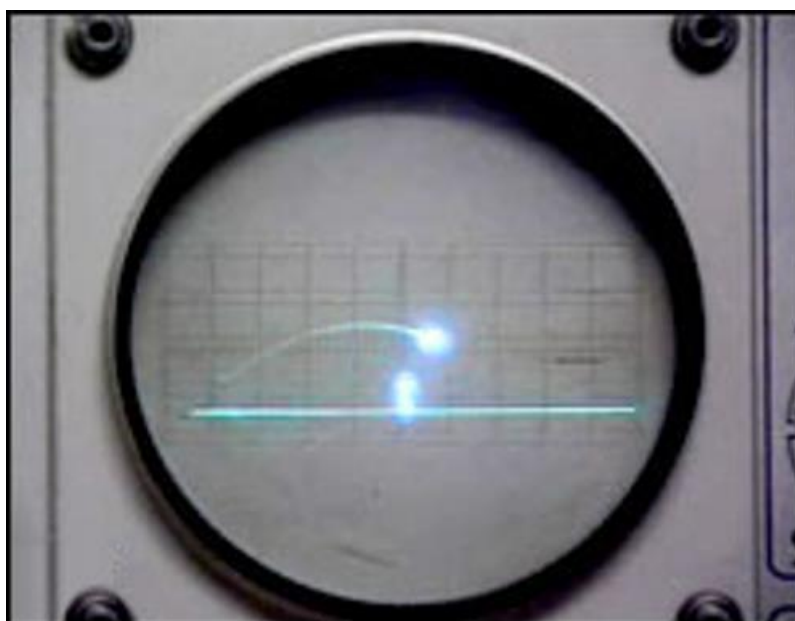


Figura 01 - *Tennis for Two*, o primeiro jogo digital.¹

¹ Disponível em: <http://gamehall.uol.com.br/v10/tennis-for-two-o-primeiro-game-da-historia-completa-55-anos/>

A afinidade com a área de *Game Design* e no geral o design voltado para entretenimento foi o motivo maior por trás da decisão de se criar um jogo para o projeto de diplomação em programação visual. A produção de um *videogame*, como uma peça visual e interativa, demanda as habilidades de um designer para garantir o melhor resultado, tanto em áreas maiores como o próprio funcionamento do jogo quanto aspectos menores como os seus personagens.

A intenção do projeto *Encore* é criar um jogo utilizando as habilidades e conhecimentos adquiridos ao decorrer do curso de Design, que seja capaz de ser diferenciado de outros *videogames* semelhantes, tendo sua própria identidade e trazendo algo de novo para o seu gênero. O jogo também é um teste final para o método utilizado e a capacidade de se realizar um projeto de design.

Para alcançar os objetivos gerais, primeiro foi essencial ter a ideia geral para o jogo, incluindo o tipo de jogo que ele deve ser. Sabendo o gênero do jogo, é capaz de detectar clichês e padrões existentes em jogos semelhantes, para então definir maneiras de fugir desses clichês e criar uma experiência que torne o jogo único entre seus similares.

Por fim, o projeto *Encore*, nomeado em referência ao ato de artistas repetirem ou estenderem suas performances a pedido da audiência, tem a intenção de ser uma celebração do que foi aprendido e realizado desde o ingresso no curso de Design.

Para iniciar o relato do projeto, será descrita a metodologia utilizada ao decorrer de sua realização, então se discutirá a proposta geral do Jogo e sua primeira visualização (chamada de *High-Concept*), conceituação necessária para se entender o desenvolvimento, os requisitos para a produção, e por último o desenvolvimento do jogo em si, com suas variações e imagens do resultado.

2. METODOLOGIA

O método aplicado tem como intenção definir uma experiência alvo para o produto final desde o início. Desde a concepção da ideia do jogo, é mantido em mente quais emoções o jogo deve evocar com suas representações gráficas, sonoras e a interação com o jogador. De certa forma, o projeto já inicia com uma expectativa de qual deve ser o resultado final, mesmo que ela esteja sujeita a mudanças e alterações.

O primeiro passo na criação, para a concepção inicial, é registrar a ideia o melhor possível, o mais rápido possível. É o *High-concept*, que será explicado no capítulo a seguir. Devido ao foco no alvo final desde a concepção, essa etapa talvez seja a mais importante, pois é a primeira representação visual da experiência desejada para o jogo, portanto a primeira versão da solução. Mesmo que o produto final seja completamente diferente

Após o registro inicial, inicia a fase de pré-produção. Nessa fase, são criadas ideias sobre o funcionamento do jogo, sua história, personagens e outros elementos que irão compor o todo, e criados esboços e desenhos para a conceituação de elementos gráficos, chamados de *concept-art* (que serão explicados no capítulo de conceituação). Ao mesmo tempo, referências musicais, visuais e de mecânica são investigadas em busca de clichês a serem evitados e soluções que podem ser utilizadas, sempre levando o alvo, definido no high-concept, em consideração.

A segunda fase do desenvolvimento é a própria produção do jogo. É nela que são criadas as peças gráficas e sonoras que comporão o jogo, e a própria programação e montagem do jogo é realizada, junto com ajustes e mudanças conforme necessário.

É importante ressaltar que embora seja possível separar as ações em duas etapas quando se leva em consideração o escopo total do projeto, a linha que as separa é bastante tênue, com ações podendo ser tomadas a qualquer momento durante o desenvolvimento. Por exemplo, algumas ideias para o funcionamento do jogo surgiram durante a etapa de montagem, e boa parte dos elementos gráficos foi criada conforme necessário, também durante a montagem do jogo, em vez de tudo ter sido desenhado antes da produção em si.

Porém, para cada parte individual, o desenvolvimento segue a mesma lógica: Concepção inicial, produção do elemento, implementação do elemento no programa, teste de funcionamento e ajustes quando necessário.

3. PROPOSTA DE JOGO E HIGH-CONCEPT

Nesse capítulo, primeiro se discutirá a definição de um *High-concept*, o que é e para que serve. Após isso, a ideia geral, a proposta do jogo será discutida, acompanhada pelo *High-concept* desenvolvido especificamente para o projeto.

3.1. O que é High-Concept

Na indústria de entretenimento, *high-concept* é uma maneira de concentrar a ideia de um projeto, durante sua concepção inicial, em uma frase só. Desse modo ela tenta sintetizar o máximo possível a experiência desejada no produto final, qual é o propósito dele. Ele é normalmente utilizado para apresentar a ideia para investidores em potencial.

Por exemplo, o filme *Toy Story*, dos estúdios Pixar, pode ser resumido com o *high-concept* “O que brinquedos fazem quando não tem ninguém olhando?”. Essa é a premissa inicial do filme, e serviu de base para a sua criação antes mesmo da história estar pronta.

Para o projeto, um tipo pessoal de *high-concept* foi utilizado. Nesse modelo, em vez de tentar resumir a proposta do jogo em uma frase, o objetivo é resumi-la em uma imagem. Dessa forma não é só possível transmitir a premissa do jogo em termos de história e jogabilidade, como também é possível capturar as intenções iniciais em termos de aspecto visual e emoção com mais facilidade.

Esse tipo de *high-concept* foi utilizado de maneira eficiente nos projetos realizados durante as duas vezes que cursei a disciplina de Design de Jogos na UnB, o primeiro projeto se chamando *Terracotta*, em 2015, e o segundo *Umbra*, em 2016, para apresentar a ideia à turma em busca de possíveis integrantes para o grupo que os desenvolveria.

Figura 02 - *High-Concept* do projeto *Terracotta*



Figura 03 - Personagens existentes no produto final, demonstrando o estilo visual final.



Figura 04 - High-Concept do jogo *Umbra*

Figura 05 - Resultado final do projeto.



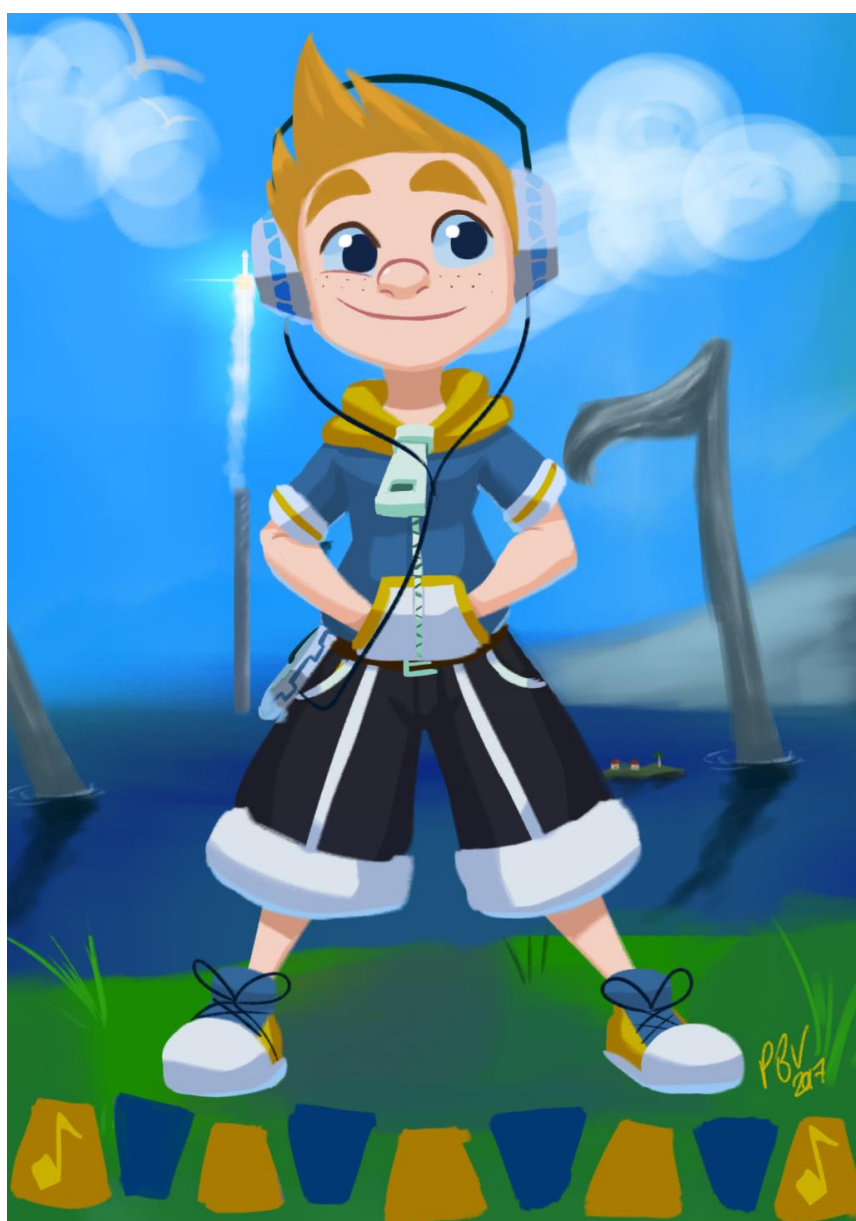
Como pode ser visto nas imagens anteriores, o *High-Concept* não representa o resultado visual final do seu projeto, porém tenta trazer o máximo possível a estética desejada. Os personagens de *Terracotta* mantém a temática inspirada por povos pré-colombianos e os tons terrosos da ilustração original, enquanto tanto o *High-Concept* quanto o resultado final do projeto *Umbra* trazem uma atmosfera sombria e opressora, e uma personagem pálida de cabelos negros que contrasta com o cenário ao seu redor.

3.2. A proposta

A ideia inicial é a criação de um jogo do gênero “plataforma e ritmo” (que será explicado no capítulo a seguir), com imagética alegre e colorida e uma música energética. A experiência desejada é que o jogador possa “se deixar levar” pela música, onde os elementos visuais do cenário intensificam a experiência musical. A intenção é tentar simular a sensação de que o seu humor e o mundo ao seu redor estão combinando com a música que você está ouvindo.

3.2.1. O *High-Concept*

Figura 06 - O high-concept do projeto



4. CONCEITUAÇÃO

Nesse capítulo, são discutidos elementos necessários para se entender o que engloba a produção de um jogo, assim como as pesquisas e referências utilizadas para definir o que o projeto deve ser.

4.1. Pesquisa e Planejamento

Aqui são descritos conceitos básicos sobre o que é jogo, as referências para o projeto e os programas a serem utilizados, etapas importantes para o planejamento do jogo.

4.1.1. O que é Jogo

Para se entender como foi feito o produto do projeto de diplomação e porque ele se configura como um jogo digital, é necessário que se compreenda o que define um jogo e o que diferencia um *videogame* de outros tipos de jogos. Em busca de um conceito claro, foram lidos textos e artigos somados à um entendimento pessoal prévio do que se define um *videogame*.

A palavra jogo provém do latim “*Jocus*”, significando brincadeira, divertimento. Jogos são atividades lúdicas que são caracterizadas por terem um conjunto de regras a serem seguidas e objetivos a serem cumpridos, e que o jogador pode vencer ou perder dependendo de sua performance. Um *videogame* se difere de jogos tradicionais principalmente por ser algo virtual. Conforme a tecnologia vai avançando, os *videogames* têm aumentado de complexidade, sendo capazes de simular mundos fictícios e criar experiências impossíveis na vida real. Porém, eles mantêm a definição comum de possuir um conjunto de regras e objetivos.

As regras se apresentam como as maneiras que o jogador pode interagir com o jogo digital, mas também se apresentam para o próprio jogo, como os limites da programação e a maneira como personagens se comportam. Porém, *videogames* modernos começaram a complicar a definição de objetivos. Vários jogos possuem objetivos diversos, com o jogador podendo escolher qual ele irá completar. Porém há também aqueles onde o objetivo é vago ou obscuro.

O importante é identificar videogames como programas virtuais, interativos, com um propósito lúdico, tendo um sistema de regras e objetivos e que o jogador possa vencer ou perder. Entendendo o que é um videogame, é possível direcionar as decisões tomadas para que o projeto seja capaz de atender a definição e ser considerado um jogo.

4.1.2. Gênero de jogo

Assim como filmes possuem gêneros, Aventura, Comédia e Romance por exemplo, jogos também os têm. Os gêneros de filme servem como categorias para separar os filmes, e dar ao espectador uma noção de que tipo de experiência ele irá receber ao assisti-los.

De igual modo, *videogames* também podem ser categorizados. Jogos de Estratégia, de Ação, de Interpretação de papéis. A diferença é que o gênero do jogo não reflete o tipo de história que ele possui, e sim a experiência no geral. Como definido anteriormente, jogos se diferem de outras mídias de entretenimento por serem interativos. Enquanto o gênero de um filme reflete a experiência ao assisti-lo, o gênero de um jogo necessariamente deve refletir a experiência de jogá-lo.

Definir o gênero do jogo o qual o projeto de diplomação se refere ajuda a nortear as decisões tomadas para a sua produção, pois já delimita um tipo de experiência desejada no produto final. Como dito no capítulo anterior projeto foi definido para pertencer ao gênero de Plataforma e Ritmo. Esse gênero é, na realidade, a mistura de dois outros e, embora já existam jogos que possam ser descritos por ele, é necessário entender os dois gêneros que o geraram.

Jogos de Plataforma são um dos tipos de jogos mais clássicos que existem. As franquias mais famosas de jogos de plataforma são parte essencial da história dos videogames, e seus protagonistas, Mario, da empresa Nintendo e Sonic, da SEGA, alcançaram popularidade comparável à de personagens como Mickey Mouse e Pernalonga.

Figura 07 - O jogo *Super Mario World* (1990), da Nintendo²



² Disponível em: <http://www.usgamer.net/articles/super-mario-world-review>

Figura 08 - O jogo *Sonic the Hedgehog* (1991), desenvolvido pela SEGA³



São jogos onde o objetivo principal é fazer um personagem ir de ponto A a B, desviando de obstáculos e inimigos e utilizando plataformas para se deslocar verticalmente. Embora cada franquia possa ter ações que as diferenciam de outros jogos (por exemplo, Sonic pode se tornar uma bolinha para se deslocar mais rápido e causar dano à seus inimigos), todos os jogos de plataforma dependem de duas ações principais: Andar/correr e Pular.

Videogames de Ritmo, como o nome do gênero sugere, enfatizam a música acima de tudo. São jogos onde o importante é entender o ritmo das músicas tocadas e ser capaz de efetuar ações de acordo com ele.

Jogos onde você deve apertar botões na hora certa, de acordo com o ritmo da música é um dos tipos mais comuns do gênero, incluindo jogos de arcade onde você deve dançar em cima de painéis com botões gigantes (*Dance Dance Revolution*), e jogos com guitarras de plástico com botões coloridos simulando o ato de dedilhar uma guitarra de verdade (*Guitar Hero*, *Rock Band*). Mas também existem jogos com outros tipos de controle, como por exemplo *Rhythm Heaven* para Nintendo DS, onde o jogador deve tocar, esfregar e riscar uma tela de toque para realizar ações diferentes ao ritmo da música.

³ Disponível em: <http://blogtectooy.com.br/sonic-the-hedgehog-nasce-um-icone-no-mundo-dos-games/>

Figura 09 - O jogo *Hatsune Miku Project Diva: Future Tone* (2016), da SEGA.



Os jogos de Plataforma e Ritmo misturam o essencial dos dois gêneros para criar uma experiência única deles. São jogos onde é necessário pular e desviar de obstáculos ao ritmo da música, sendo ele uma ferramenta vital para compreender o tempo necessário para realizar as ações.

Para que o projeto seja capaz de se encaixar nesse gênero de jogo, então, maneiras de atrelar os diversos elementos que o compõem à música presente nele deveriam ser criadas. O relatório discutirá as maneiras que o jogo se associa ao ritmo ao descrever o desenvolvimento e o resultado.

4.1.3. O que é Mecânica

As mecânicas de um jogo são elementos do seu sistema, partes que determinam a maneira que ele funciona e interage com o jogador. Por exemplo, o modo como ele controla o personagem principal e as diversas ações que ele é capaz de executar são mecânicas, assim como coisas como pontuação e obstáculos com propriedades diferentes também podem ser considerados mecânicas do jogo.

Por exemplo, um dos jogos de plataforma mais conhecidos, *Super Mario Bros*, desenvolvido pela Nintendo em 1985, para seu console *Nintendo Entertainment System*, tem as seguintes mecânicas:

O jogador pode mover o protagonista Mario horizontalmente apertando dois botões no controle, um para a direita e outro para a esquerda. Ao apertar um outro botão, o protagonista pula. Há oponentes no cenário, e encostar neles mata o personagem, obrigando o jogador a começar a fase do início, e diminui a quantidade de vidas que ele tem. Quando as vidas chegam em zero, o jogo acaba e o jogador perde. A principal maneira de derrotar os inimigos é pular em cima deles, sendo a única maneira de encostar neles sem ser danificado.

O jogador precisa completar cada fase dentro de um limite de tempo, mostrado por um contador no canto superior da tela. Na fase, ele pode coletar moedas que aumentam sua pontuação, e coletando 100 moedas o jogador recebe uma vida extra. Também há itens que fazem o personagem aumentar de tamanho, o que permite que o jogador encoste em um inimigo uma única vez sem morrer, em vez disso retornando para o tamanho inicial, e itens que deixam o personagem atirar projéteis com outro botão.

Para terminar a fase, o jogador deve mover para a direita até encontrar uma bandeira que representa o final da fase, desviando dos inimigos e de outros obstáculos, como buracos que devem ser pulados. Dependendo da altura em que o jogador encosta na bandeira final, ele recebe um bônus maior. Após terminar a fase, se inicia a próxima.

Figura 10 - O jogo *Super Mario Bros.* (1985), da Nintendo.⁵



Grande parte dos jogos de plataforma possuem uma mecânica bem próxima a delimitada pelo jogo descrito, com variações que os tornam únicos, sendo eles mais complexos ou mais simples. Porém, todos eles possuem em comum o deslocamento horizontal e o pulo.

Os jogos de plataforma e ritmo vão associar essas mecânicas básicas de plataforma à um ritmo musical, adicionando um elemento temporal ao deslocamento, além da contagem regressiva. Nesse gênero, fatores como a hora certa de se executar uma ação entram em vigor.

Tendo noção do que se refere o relatório ao falar sobre mecânica, é possível realizar uma análise de jogos similares e descrever mecânicas que parecem desejáveis ou não para o projeto final, além de poder discorrer as mecânicas presentes na hora de discutir o funcionamento do jogo final.

⁵ Disponível em: <http://uk.businessinsider.com/super-mario-run-shigeru-miyamoto-original-creators-2016-9>

4.1.4. Referências de Mecânica

A intenção do projeto é criar um jogo de plataforma e ritmo que se diferencie dos seus similares e tenha uma identidade própria. Para isso é necessário analisar jogos do mesmo gênero e como eles funcionam, na tentativa de encontrar padrões e clichês que possam ser quebrados, e ideias que possam ser mantidas.

4.1.4.1. *The Impossible Game (2009), por FlukeDude, e Geometry Dash (2013), por RobTop Games*

Figura 11 - *The Impossible Game*⁶



Embora sejam dois jogos diferentes, esses jogos de aspecto visual simples e geométrico são quase idênticos em funcionamento e apresentação, portanto serão descritos juntos.

São jogos de plataforma onde o personagem jogável, um quadrado, já move horizontalmente por conta própria, e o jogador tem apenas o controle sobre o ato de

⁶ Imagem adquirida ao jogar o jogo,

pular. Qualquer obstáculo que faça o quadrado parar de andar faz com que o jogador perca o desafio e tenha que recomeçar do zero.

Se categorizam como jogos de plataforma e ritmo pois os obstáculos de cada fase estão posicionados de acordo com a música, e o ritmo da música indica o momento certo em que a ação de pular deve ser executada para que o jogador consiga alcançar a próxima plataforma com sucesso.

Figura 12 - *Geometry Dash*⁷



Os jogos se diferem, pois, Geometry Dash introduz um segundo segmento ao gameplay, onde o jogador toma controle de uma nave capaz de voar. O botão que executa a ação de pular agora faz com que essa nave pegue impulso e suba, independente se ela está encostando no chão ou não, e não o apertar faz com que a nave desça. Nesta seção, o jogo passa a ser sobre conseguir manter a nave na altitude certa, sem encostar nos obstáculos abaixo ou acima dela. O ritmo então se torna a frequência em que o botão deve ser apertado para manter uma altitude estável.

⁷ Imagem adquirida ao jogar o jogo.

4.1.4.2. *Bit.Trip Runner* (2010) por Choice Provisions.

Figura 13 - *Bit.Trip Runner*⁸



Bit.Trip Runner mantém muitas das convenções dos jogos já analisados, porém se diferencia na apresentação mais complexa e tendo uma mecânica mais desenvolvida. Além de ter o botão de pular, o jogador também tem acesso a um botão que pode fazer o protagonista agachar, para desviar de obstáculos que não podem ser desviados com um pulo.

A mecânica mais interessante de Bit.Trip Runner é o modo como ele implementa a música. O ritmo ainda é essencial para a execução correta dos pulos e agachamentos, porém a música se transforma em um feedback positivo. A melodia se dá quando o jogador executa uma ação corretamente, seja ela desviar de um obstáculo ou pegar um item. Além disso, há itens especiais que, ao serem coletados, tornam a música mais detalhada e empolgante, e aumentam a quantidade de informação na tela conforme são pegos, introduzindo um rastro de arco-íris ao protagonista ao alcançar o nível máximo.

⁸ Disponível em: <http://www.ign.com/articles/2011/03/21/bittrip-runner-review>

Conforme a análise desses três jogos, nota-se um certo padrão no gênero de Plataforma e Ritmo. São normalmente jogos onde o movimento na horizontal é automático, e o desafio está em ter um reflexo rápido o suficiente para desviar dos obstáculos de acordo com a música. Dessa maneira, pode-se dizer que nesse estilo de jogo, o jogador é quem está sujeito ao ritmo da música, mantendo-se no ritmo para completar o objetivo de alcançar o final de cada fase. Na tentativa de dar um diferencial para o jogo, foi decidido seguir por outra direção, tornar o cenário dependente do ritmo da música, enquanto o jogador tem controle livre sobre o protagonista.

Para melhor exemplificar esse desejo, segue um exemplo encontrado em uma fase de um outro jogo. Uma versão mais simples do tipo de associação entre cenário e música desejado.

4.1.4.3. *Super Mario 3D World (2013), Nintendo*

Figuras 14 e 15 - Plataformas que alternam entre si de acordo com o ritmo da música⁹



Sendo um dos jogos mais modernos da franquia Super Mario, é um jogo tridimensional que tenta imitar a mesma fórmula dos jogos antigos, ou seja, mesmo com um deslocamento em três dimensões, o desafio ainda é chegar até o final da fase

⁹ Disponível em: <https://www.gamerguides.com/super-mario-3d-world/walkthrough/world-flower/world-flower-8-blast-block-skyway>

dentro do limite do tempo, pulando e evitando quedas para fora da área da fase entre outros obstáculos.

As fases “*Bleep Block Skyway*” e “*Blast Block Skyway*” possuem uma mecânica central, elas têm plataformas coloridas que alternam entre si de acordo com o ritmo da música. Funcionam da seguinte forma: Existem plataformas da cor azul e da cor vermelha, e apenas uma cor é exibida de cada vez. Depois de um certo tempo, fácil de ser delimitado pela música, as plataformas se alternam, com metade se ocultando e outra metade aparecendo, conforme mostrado na imagem acima.

Ter uma noção do ritmo da música de fundo e da fase é essencial para ser capaz de completar o objetivo sem apelar pela sorte, portanto, as fases analisadas podem ser consideradas fases de plataforma e ritmo, onde os obstáculos e o próprio layout da fase se alteram de acordo com a música enquanto o jogador ainda tem liberdade para se locomover conforme desejado.

As fases são capazes de representar o tipo de experiência desejada para o jogador, embora elas demonstrem tal mecânica com apenas um tipo, o dos blocos alternados. Cabe ao projeto inventar outras maneiras de trazer esse aspecto de um cenário atrelado ao ritmo da música, além de implementações diversas desse mesmo ritmo, em vez de uma frequência uniforme através da fase inteira.

Entender quais são as referências positivas e negativas para o que virá a ser a mecânica do jogo trará clareza na hora de explicar as decisões tomadas para o funcionamento, e principalmente deve ser levado em consideração como prova de que o jogo se enquadra no gênero de Plataforma e Ritmo mesmo quebrando os padrões determinados por outros jogos do gênero.

4.1.5. Referências Visuais

Enquanto mídias audiovisuais interativas, os jogos dependem de seu aspecto visual para transmitir a experiência desejada, e também como atrativo para o jogador. Uma das partes essenciais para garantir a imersão no mundo do jogo é uma estética que seja consistente com o tema, atraente e que, principalmente, possua clareza para que seja um apoio para o jogador ser capaz de entender o jogo.

Na busca de um estilo visual para o projeto, foi necessário pesquisar outros *videogames* já existentes no mercado, para ser capaz de criar algo único. Como na pesquisa de mecânica, inicialmente foram analisados jogos do mesmo gênero, com o objetivo de se distanciar de padrões já existentes. A partir de então, foi gerada uma noção de como o jogo poderia ser, e pesquisados videogames que tenham uma estética parecida com a desejada.

Os jogos de ritmo e plataforma analisados foram os mesmos que tiveram sua mecânica descrita no tópico anterior. Como pode ver pelas imagens no capítulo anterior, eles possuem algumas semelhanças que indicam uma preferência por temas e elementos visuais em comum, mesmo quando a execução é diferente.

Normalmente jogos de plataforma e ritmo têm preferido um aspecto visual simples, minimalista e muitas vezes com aspectos de luzes neon e temática futurista. Os jogos mais comuns do gênero, que podem ser jogados em browsers, como *The Impossible Game* e *Geometry Dash*, não possuem uma identidade muito delimitada, preferindo utilizar formas geométricas simples em sua composição.

Na tentativa de diferenciar o projeto de seus similares, foi decidido buscar um estilo visual bem diferente. Em vez de tomar a direção do futurismo e do neon, focar em temas de fantasia, cores vibrantes, mas naturais e um aspecto infantil como o de um desenho animado. Substitui-se os cenários tecnológicos e com luzes neon por cenários naturais como montanhas, praias e planícies, com casas de telhas vermelhas.

Tentando melhor visualizar a estética desejada, foram pesquisados jogos de diversos gêneros, boa parte sem nenhuma relação com o projeto em termos de mecânica. Eles serão descritos a seguir, explicando que tipo de jogo eles são e quais partes de seu aspecto visual é interessante de ser adaptada e incorporada ao projeto.

4.1.5.1. *Jak and Daxter: The Precursor Legacy* (2001), de Naughty Dog.

Figura 16 - Uma cena do jogo¹⁰



Jak and Daxter ficou conhecido por ser um jogo de aventura que permitia que o jogador atravessasse todas as regiões de seu mundo sem precisar esperar que o sistema as carregasse separadamente, ou seja, ele não tinha uma divisão tradicional de fases e regiões.

Possui um estilo visual bastante colorido e com animações exageradas e expressivas, e se passa em um local com uma temática parecida com a almejada, com planícies, praias e casas de telhas vermelhas.

¹⁰ Disponível em: https://store.playstation.com/en-us/product/UP9000-CUSA02522_00-SCUS971240000001

4.1.5.2. *Franquia Kingdom Hearts (2002-2018), por Square-Enix*

Figura 17 - Os protagonistas da franquia, Sora, Pato Donald e Pateta.



11

Os jogos da franquia *Kingdom Hearts* são jogos de ação e aventura que focam em combate rápido e visualmente estimulante, e na exploração do cenário. Possui uma premissa única, sendo a história de um garoto que viaja através de mundos baseados em filmes da Disney, contando com a ajuda do Pato Donald e do Pateta, personagens clássicos.

Devido ao seu tema, a sua direção de arte tenta ser o mais próximo possível dos filmes clássicos que estão sendo representados no jogo. Para isso, ele dá um aspecto que remete ilustrações feitas à mão para seus cenários e personagens, tentando imitar os filmes o melhor possível.

Além disso, um elemento bastante presente no design de seu “universo” é a mistura entre fantasia e tecnologia, e as vestimentas de seus protagonistas reflete

¹¹ Disponível em: <https://nowloading.co/p/the-best-order-to-play-the-games-in-kingdom-hearts-1-2-hd-remix-ps4/4230265>

isso. É um jogo onde é possível ver personagens de contos de fadas e filmes clássicos da Disney interagindo com computadores, ambientes virtuais e outros tipos de tecnologia que chegam a ser futuristas.

Figura 18 - O clássico salão de dança do filme *A Bela e a Fera* (1991), representado no jogo¹² *Kingdom Hearts II* (2005)



4.1.5.3. *Yooka-Laylee* (2017), de Playtonic Games

Figura 19 - Um dos cenários do jogo¹³



¹² Disponível em: https://www.models-resource.com/playstation_2/kh2fm/model/1602/

¹³ Disponível em: <https://www.polygon.com/2017/4/4/15171316/yooka-laylee-review-xbox-one-pc-playstation-4>

Yooka-Laylee é um jogo de plataforma estilo *collect-a-thon*, ou seja, que foca na busca e coleta de vários objetos espalhados pelo cenário. Foi desenvolvido por um estúdio composto de ex-membros da empresa *Rareware*, conhecida por ter feito vários jogos do mesmo gênero nos anos 90. De certa maneira, é considerado um retorno da fórmula após quase duas décadas.

Tem um aspecto visual colorido e personagens bastante estilizados, porém o mais marcante desse jogo, em termos de referência visual para o projeto, é o modo como ele estiliza o cenário. Pedras retangulares, símbolos que remetem à uma estética tribal (como totens espalhados pelo cenário) e formações geográficas nada realistas.

Todos os jogos analisados têm elementos visuais coloridos e os personagens semelhantes a desenhos animados infantis, mas cada um possui alguma peculiaridade que pode ser incorporada e adaptada ao projeto. Porém, é vital que o videogame a ser projetado também se diferencie consideravelmente deles em termos de representação visual e estilo artístico, podendo compartilhar temas mas tendo suas próprias diferenças, sua própria identidade. Mais adiante, quando for a hora de definir os requisitos do estilo visual do jogo, não só devem ser levadas em consideração as semelhanças como também as diferenças que o projeto terá em relação aos jogos analisados.

4.1.6. Referências Musicais

A música de um jogo possui um papel essencial pois, como a trilha sonora de um filme, serve para ambientar o jogador no mundo do jogo, e transmitir uma emoção desejada. Por exemplo, o jogo *Persona 5*, lançado em 2017 e desenvolvido pela empresa japonesa ATLUS, utiliza suas músicas para transmitir ao jogador a sensação de rebeldia e insubordinação.

Existem jogos que tocam suas músicas repetidamente, e jogos que tocam as músicas de maneira esporádica. Por exemplo, jogos que preferem focar nos sons do cenário para ambientar o jogador no mundo, normalmente utilizam música apenas em momentos importantes da história ou como uma quebra da monotonia dos efeitos sonoros.

Porém, não existem gêneros de jogo que dependam mais de música do que os jogos musicais, como o nome indica, e consequentemente os jogos de Plataforma e Ritmo. A experiência do jogador é diretamente associada à música tocada durante o jogo, assim como o próprio desafio do jogo depende de seu ritmo para ser resolvido.

A música do jogo deve refletir o tema do jogo. Ela deve ter um ritmo forte e evidente pois o jogo depende do ritmo para se enquadrar no gênero definido, e também deve agir em conjunto com os elementos visuais do jogo para transmitir uma sensação desejada.

Em busca de um estilo musical para o projeto, foram buscadas diversas músicas como referência, e separadas as que mais se enquadravam na experiência final desejada. Foi definido que o jogo deveria ter músicas eletrônicas (utilizando sintetizadores) e que tragam uma sensação alegre para o jogador. Para melhor entender o estilo musical que o jogo deveria ter, foram encontradas as seguintes músicas:

- a) NAKATA, Yasutaka. **Crazy Crazy**. 2017.
- b) HYPER POTIONS. **Friends**. 2017.
- c) MITCHIE M. **Ai Dee**. 2012.
- d) MODJO. **Lady (Hear me Tonight)**. 2001. Versão “#FITNESS VIP” criada por DJ DEBUSSEY TURNPIKE VS GHIBLI em 2014.
- e) TAKEMURA, Kiriko. **Harajuku Iyaho!**. 2017.
- f) HARRISON. **Happy Boy**. 2014.

Das músicas listadas, *Crazy Crazy* e *Happy Boy* são as mais próximas do desejado para o projeto. Ambas possuem um aspecto divertido e quase infantil, tem uma marcação forte de ritmo, e instrumentação eletrônica. Elas têm menos intensidade comparadas com as outras músicas da lista, sendo menos frenéticas e abririam espaço para uma exploração mais calma e devagar do cenário do jogo. Porém, ao mesmo tempo, continuam sendo animadas e alegres. Músicas parecidas com as outras da lista seriam mais adequadas para jogos com uma velocidade maior e mais tensão.

As referências musicais são tão vitais quanto as referências visuais para a produção do jogo, talvez até mais. As decisões tomadas na pesquisa serão levadas em consideração na hora de produzir a parte sonora do jogo.

4.1.7. Pesquisa de Ferramentas

A produção de um jogo depende de várias ferramentas para a criação de suas várias partes. Os elementos visuais vão necessitar de programas que sejam capazes de produzi-los, sejam programas de desenho e animação ou de manipulação digital. Os elementos de programação podem utilizar de softwares que facilitam o processo de produção do jogo e também aumentam a compatibilidade com diferentes tipos de plataforma, ao garantir uma estabilidade maior. São as chamadas *Engines* (motores) de jogo.

Como cada programa possui suas vantagens e desvantagens e pode oferecer resultados diferentes, foi feita uma leve comparação entre diversas alternativas, listando os pontos positivos e os negativos. Foram pesquisados programas para a criação de elementos visuais e programas para a montagem do jogo. Não foram pesquisados programas para a produção de elementos sonoros e musicais por motivos que serão discutidos no capítulo da geração de alternativas.

Os programas para a criação de imagens pesquisados foram medidos em termos de acessibilidade e preço, facilidade de uso e experiência, e o resultado final. Todos tiveram um requisito mínimo, ser capaz de gerar animações, uma vez que o projeto do jogo depende de animações.

4.1.7.1. Adobe Photoshop CS6

Figura 20 - Alguns painéis do programa¹⁴



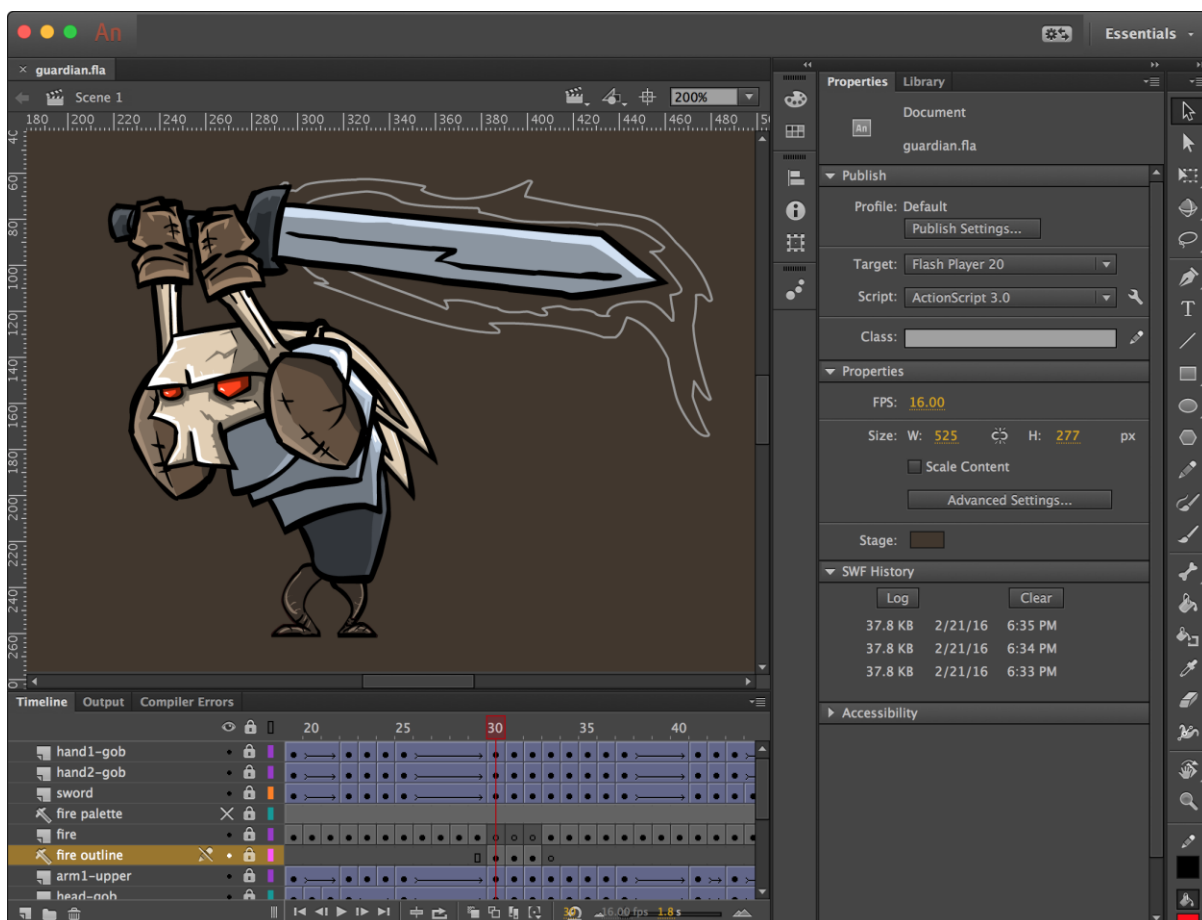
O primeiro programa a ser analisado, por ter maior familiaridade com o seu uso. É voltado para a modificação de imagens e fotos, como a criação de montagens, porém também é utilizado como uma robusta ferramenta para a pintura digital, sendo capaz de simular vários pincéis e ter bastante customização.

As vantagens desse *software* são a própria familiaridade e acessibilidade, permitindo a criação de desenhos e pinturas com mais facilidade, velocidade e um resultado melhor. Porém, embora o programa suporte a criação de diversos tipos de animação, o seu sistema para animar é muitas vezes confuso e ineficiente, comparado com programas voltados para a animação, sendo essa a maior desvantagem.

¹⁴ Disponível em: <https://www.austinfilm.org/event/intro-photoshop-dec-2017/>

4.1.7.2. Adobe Animate

Figura 21 - O Animate¹⁵



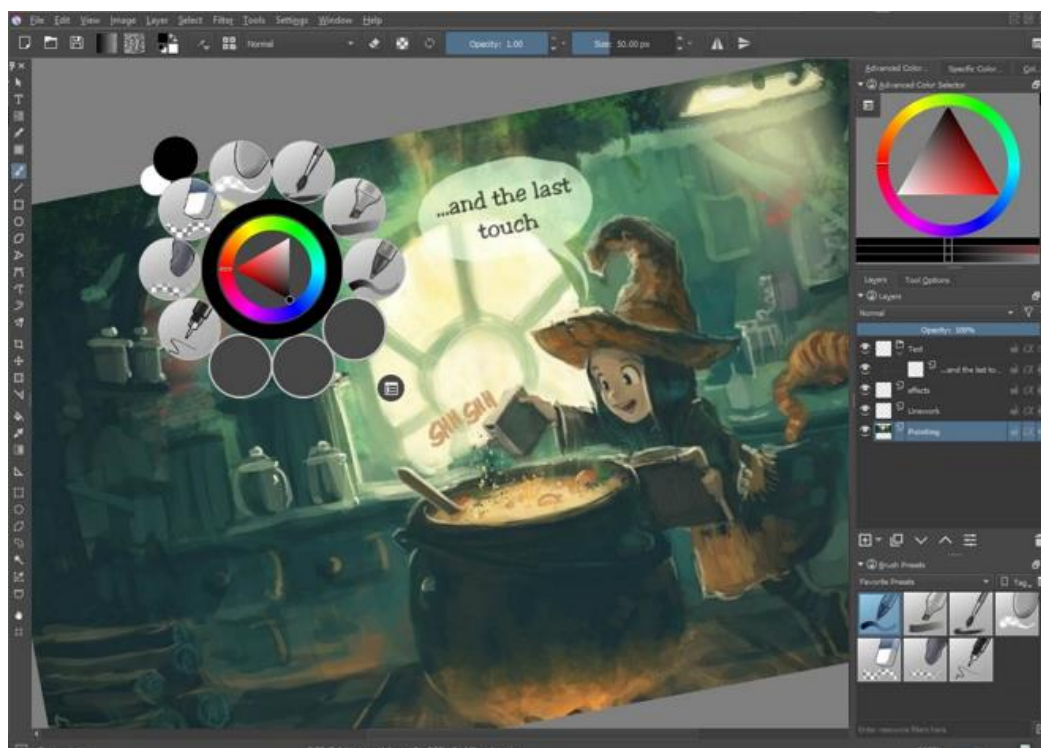
O programa *Adobe Animate* é um sucessor do *Adobe Flash*, lançado conforme a própria plataforma de animação anterior se tornou obsoleta em favor da plataforma HTML5. Tem um funcionamento quase idêntico ao seu antecessor, porém se torna compatível com HTML5 e tem novas ferramentas.

Por ter aprendido a animar pelo Flash antes de qualquer outra plataforma, as ferramentas do Animate são práticas de se utilizar, e é fácil de se criar animações, permitindo um controle maior do que o *Adobe Photoshop*. Porém, é um programa limitado à imagens vetoriais, de cores chapadas, o que limita bastante o potencial para o estilo visual do jogo.

¹⁵ Disponível em: <http://www.mrbradshaw.ca/animation/2d-animation>

4.1.7.3. Krita

Figura 22 - Um desenho sendo realizado no Krita¹⁶



Krita é uma plataforma completamente grátis voltada para a arte digital. Possui várias ferramentas que simulam diversos tipos de material, e atalhos voltados para facilitar o uso do artista. Também possui ferramentas que ajudam o desenvolvimento de peças para jogos, tendo um método de criação de padrões e módulos de maneira mais fácil, facilitando a criação de cenários modulares. Em suas últimas versões, vêm com o seu próprio sistema para animação tradicional, tentando ser mais prático que o *Adobe Photoshop* sem perder a qualidade do desenho.

Infelizmente, programas diferentes possuem peculiaridades diferentes, assim como diferentes pincéis produzem resultados diferentes, dependendo do tipo de pêlo utilizado, mesmo sendo de tamanhos semelhantes. Krita é capaz de produzir resultados ótimos, porém o tempo para se acostumar com o programa eliminaria qualquer tempo poupado que o sistema de animação simplificado e melhorado seria capaz de oferecer.

¹⁶ Disponível em: <https://krita.org/en/features/highlights/>

4.1.7.4. Engine

Para a *Engine* do jogo, o foco foi em encontrar um meio-termo entre facilidade de uso e liberdade de criação. Muitas vezes, há programas capazes de criar jogos sem o desenvolvedor precisar escrever uma única linha de código, apenas utilizando das próprias ferramentas dos programas. Porém, esses mesmos programas acabam sendo bastante limitados, criando jogos simples e que rodam em muito poucas plataformas.

Foi criado um quadro, listando a quantidade de programação necessária, as limitações e as plataformas compatíveis. Como pode ser visto abaixo, a relação entre a simplicidade de uso e as limitações é inversa.

Quadro 1 - Comparação entre os softwares analisados.

Nome do Software	Quantidade de Programação Necessária	Limitações	Plataformas Compatíveis
Unity	Depende de programação para montar o jogo.	Quase nenhuma, possível fazer o jogo do jeito desejado.	Compatível com mais de 25 plataformas, incluindo PC e Mac, celulares e todos os consoles atuais.
GameMaker Studio	Não depende de conhecimentos em programação para montar o jogo, porém permite utilizar código para funções avançadas.	Possível fazer o jogo do jeito desejado, portanto que seja um jogo bidimensional.	Não tão compatível quanto o Unity, porém oferece um bom leque de plataformas para lançar o jogo.
GDevelop	Uso fácil, sem precisar programar, utilizando um sistema de clicar e arrastar peças para montar o jogo.	Bem mais limitado que os anteriores, porém ainda capaz de executar o mínimo necessário para o jogo.	Exporta apenas como executável para Windows e como HTML5.
Construct2	Também com um sistema de clicar e arrastar, porém mais simples de usar que o anterior.	Bastante limitado.	Apenas HTML5.

Após a análise, foi realizada a decisão de utilizar o *Adobe Photoshop CS6* para a criação de elementos gráficos, devido à sua familiaridade, enquanto para a montagem do jogo foi preferido escolher entre o *GameMaker Studio 2* e o *GDevelop*, por estarem mais próximos de um meio-termo entre as relações mencionadas anteriormente. Inicialmente será utilizado o *GameMaker Studio 2*, e se ele se provar complexo demais para o limite de tempo, o desenvolvimento passará para o *GDevelop*.

4.1.8. Tipos de animação

Jogos enquanto mídias visuais envolvem algum tipo de movimento, logo envolvem a produção de animações. Com o avanço da tecnologia, surgiram várias possibilidades para o universo dos *videogames*, desde as simples imagens de baixa resolução e complexidade da década de 80 até os gráficos 3D avançados que chegam cada vez mais perto da vida real, cada um tendo seu próprio nível de complexidade e oferecendo um resultado diferente para o jogo.

Definir que tipo de animação seria feita para o projeto é essencial, pois essa parte é capaz de mudar completamente não só o estilo visual do projeto como também os requisitos para o executar e o próprio funcionamento do projeto. Um jogo 2D e um jogo 3D funcionam de maneiras diferentes. Cada um necessita de uma mentalidade diferente e maneiras diferentes de se aproximar um problema.

Para ser capaz de decidir qual direção tomar, foi feita uma análise de vários tipos de animação conhecidos para jogos, pensando nas vantagens e desvantagens. Foram levados em consideração o tempo e esforço que cada um demandaria e o resultado final que cada um oferece. Cada um deles será descrito e analisado nos parágrafos a seguir.

4.1.8.1. Tradicional

A animação 2D tradicional foi o primeiro tipo de animação a existir. É o que foi utilizado desde o início do século XX para se criar filmes de curta e longa-metragem por grandes estúdios como a Disney e a Warner Bros., antes do surgimento de animações em 3 dimensões.

Se parte do princípio que filmes, ou seja, imagens em movimento, se formam a partir de uma sequência de imagens sendo executada em uma velocidade que a passagem de uma imagem a outra não seja perceptível. Em animação, significa que cada uma dessas imagens ou, em termos de vídeo, cada um dos quadros é desenhado um por um em separado, formando um todo ao serem vistos em sequência.

Figura 22 - Ciclo de caminhada, animação tradicional. ¹⁷



Utilizar esse método de maneira pura não é muito comum em *videogames*, pois demanda bastante tempo e habilidades que outros tipos de animação talvez não precisem. Porém, utilizar esse método possibilita um aspecto visual com maior nível de detalhamento e é capaz de oferecer um resultado mais próximo do desejado durante a pré-produção.

4.1.8.2. Pixel-art

Um Pixel é a menor unidade visual em uma tela digital. No início da era virtual, quando computadores e jogos não tinham muita capacidade gráfica e não existiam monitores HD, os elementos gráficos dos jogos eram simples e de baixa resolução. Um estilo visual inspirado nessa estética passou a ser chamado *Pixel art*.

¹⁷ Disponível em: angryanimator.com

Os gráficos dos jogos até a metade da década de 90 eram em sua maioria 2D e com uma resolução baixa. Os pixels eram evidentes naquela época, e os jogos eram feitos com os pixels em mente.

Figura 23 - *Super Mario World* (1990), Nintendo.



Hoje em dia, gráficos em *Pixel Art* ainda existem, principalmente no cenário de jogos independentes. Muitos tentam capturar uma sensação de nostalgia ao utilizar os mesmos gráficos que eram usados nas décadas de 80 e de 90, e por serem estilos visuais com pouco detalhamento, são mais fáceis de se criar.

O método é bastante parecido com a animação tradicional, sendo feito a partir de quadros individuais. Porém, devido à simplicidade visual, é mais rápido de ser feito e exige menos detalhes. Infelizmente, a praticidade muitas vezes é às custas do detalhamento visual, obrigando o jogo a ter uma estética mais simples. Criar algo detalhado por meio de *Pixel Art* é possível, porém, como é uma técnica comparável ao pontilhismo, onde cada pixel é colocado individualmente, algo bastante detalhado possivelmente demanda mais tempo do que desenhar ou pintar normalmente.

Figura 24 - *Sonic Mania* (2017), da SEGA



4.1.8.3. “Marionete” ou *In-between*

Um método bastante usado em desenhos animados criados na década de 2000 por ser rápido e barato, o método de *in-betweening* se trata de dividir o personagem em partes como um boneco articulado, e animar as partes como um manequim. São determinados o ponto inicial e o ponto final de um movimento, e o computador calcula o resto automaticamente.

Figura 25 - *Inbetweening* no Adobe After Effects¹⁸



Há como fazer animações de qualidade com esse método, porém isso demanda tempo e habilidade. A maioria das animações feitas com *in-betweening* possuem um aspecto bastante artificial, podendo dar um aspecto ruim ao projeto, ou não. Dependendo do projeto, o estilo de animação complementa decentemente com o tema, permitindo uma suavidade difícil de se obter com a animação tradicional.

Figura 26 - O jogo *Broken Age* (2015), de Tim Schafer.



¹⁸ Disponível em: surfacedstudio.com

4.2. Pré-produção

A pré-produção é a criação de elementos que ajudam a conceituar o produto final. São peças gráficas e conceitos criados para o produto, mas que não necessariamente fazem parte da versão final. Em vez disso, são utilizados como referência e até como “manual” para a produção dos elementos finais.

4.2.1. Concept-art

Concept Art (“arte conceitual” em português), é um tipo de pré-produção vital para vários tipos de mídia visual, incluindo animação e jogos. O termo começou a ser utilizado por volta dos anos 30 pelo estúdio Walt Disney e é uma parte essencial para o design de personagens, cenários e qualquer elemento que exista dentro do produto. É basicamente a ideia por trás dos esboços e da geração de alternativas existente no âmbito do design, porém voltado para a geração de personagens, cenários e objetos que farão parte de um todo.

Figura 27 - *Concept Art* de uma criatura chamada Hipogrifo, para o filme *Harry Potter e o Prisioneiro de Azkaban* (2003), dos estúdios Warner Bros.¹⁹



¹⁹ Disponível em: <https://nerdist.com/gallery-rare-concept-art-courtesy-of-a-new-harry-potter-book-the-creature-vault/>

Como um esboço, *Concept Art* não é criada com o propósito de ser utilizada diretamente no produto final, e sim como uma base de referência. É uma etapa essencial por ser a visualização inicial de como elementos deverão ser apresentados no produto final. Supondo que a produção de cada personagem, cada cenário e cada item é um pequeno projeto de design gráfico por si só, então *concept arts* são quaisquer peças visuais produzidas antes de se chegar em um produto final, e também um tipo de manual da identidade visual daquele item em questão.

A criação de concept art é essencial para a produção do jogo, pois é a partir dela que a representação visual final do jogo será adquirido, sendo uma etapa de planejamento para todas as suas partes. As peças de *concept art* criadas para o jogo serão analisadas quando for se tratar de geração de alternativas.

Figura 28 - Um tipo de *Concept Art* para o filme Frozen (2013), de *Walt Disney*, destrinchando possíveis expressões faciais de uma das protagonistas.²⁰



²⁰ Disponível em: <http://time.com/3672324/frozen-concept-art/>

4.2.2. Enredo

Jogos são mídias interativas, onde o jogador tem controle sobre a progressão da história, dando um aspecto não linear até para jogos que possuem uma sequência linear de eventos, de fases. Isso não impede que a maioria dos jogos tenham algum tipo de trama, premissa e história, por mais simples que sejam.

A trama de um jogo pode ser longa e bem-desenvolvida, como pode ser visto em jogos do gênero RPG (jogo de interpretação de papéis), ou extremamente curta e simples, ficando em segundo plano. Porém, mesmo a menor e mais surreal premissa é importante para um jogo que tenha personagens pois é ela que determina o contexto em que o jogador se encontra, o objetivo que ele deve cumprir e a motivação que ele tem para cumprí-lo.

Por exemplo, a trama do primeiro jogo da franquia Super Mario, *Super Mario Bros.*, se trata do protagonista, Mario, ter que alcançar e salvar a princesa Peach, que foi sequestrada pelo rei monstro Bowser e seu exército de monstros. A partir de então, o jogador tem uma razão para o que ele está fazendo, um contexto.

Por ser um jogo com personagens, criar uma trama para o projeto se tornou essencial, para justificar os objetivos do jogo e dar uma motivação para o jogador seguir em frente. Embora seja possível criar uma história inteira e bem-desenvolvida para o jogo com facilidade, ou seja, com vários eventos e pontos na trama, ao decorrer do jogo inteiro, foi preferível criar uma história que possa ser compreendida e resumida apenas em uma fase do jogo.

A história será discutida em detalhes junto com as *concept-arts*, pois a história influencia o desenvolvimento visual do jogo e vice-versa.

4.3. Produção

Os termos descritos nesse tópico são utilizados para os elementos do produto final em si. São objetos que serão aplicados e utilizados no jogo, em vez de peças de conceptualização.

4.3.1. Assets

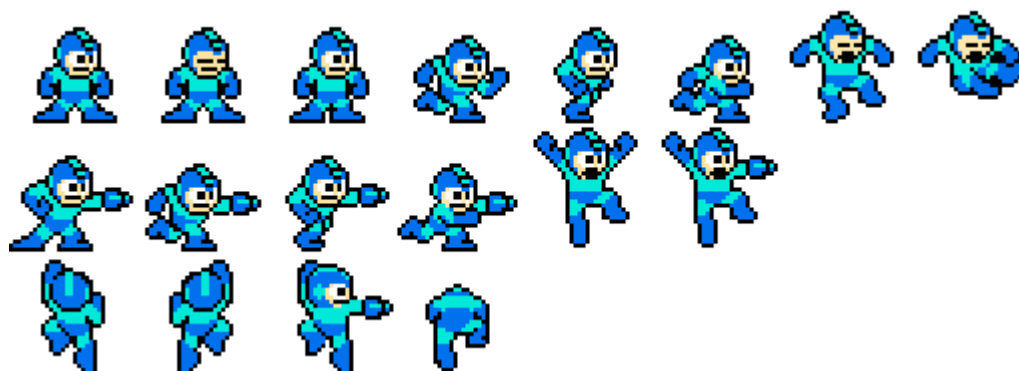
Asset é uma palavra em inglês que significa alguma qualidade, coisa ou pessoa útil em uma situação. Basicamente, significa um recurso. Em termos de jogos digitais, um *Asset*, ou um recurso, é algum elemento que pode ser usado para a montagem do jogo, como uma peça de cenário, um personagem, um efeito sonoro etc. No âmbito do relatório, *Assets* serão os arquivos de imagem e de música que farão parte do produto final. Se diferem de *concept-art* pois são parte do produto final em vez de servirem apenas como conceituação.

Em termos de gráficos, há dois tipos de *Asset* que são necessários de entender para o projeto, esses sendo nomeados *Sprite-sheets* e *Tilesets*. Há outros tipos de imagens em jogos, porém eles não possuem um formato característico como os tipos mencionados, sendo apenas imagens comuns.

Os *sprite-sheets* são o modo como objetos animados são importados para o jogo. Supondo que a animação do jogo tenha sido de maneira tradicional, como descrito no tópico sobre tipos de animação, então ela é composta de múltiplos quadros individuais.

Esses quadros, chamados de *Sprites* nos jogos, são organizados em uma grade onde todas as células têm tamanhos iguais, se tornando uma única imagem, conhecida como a *Sprite-sheet*.

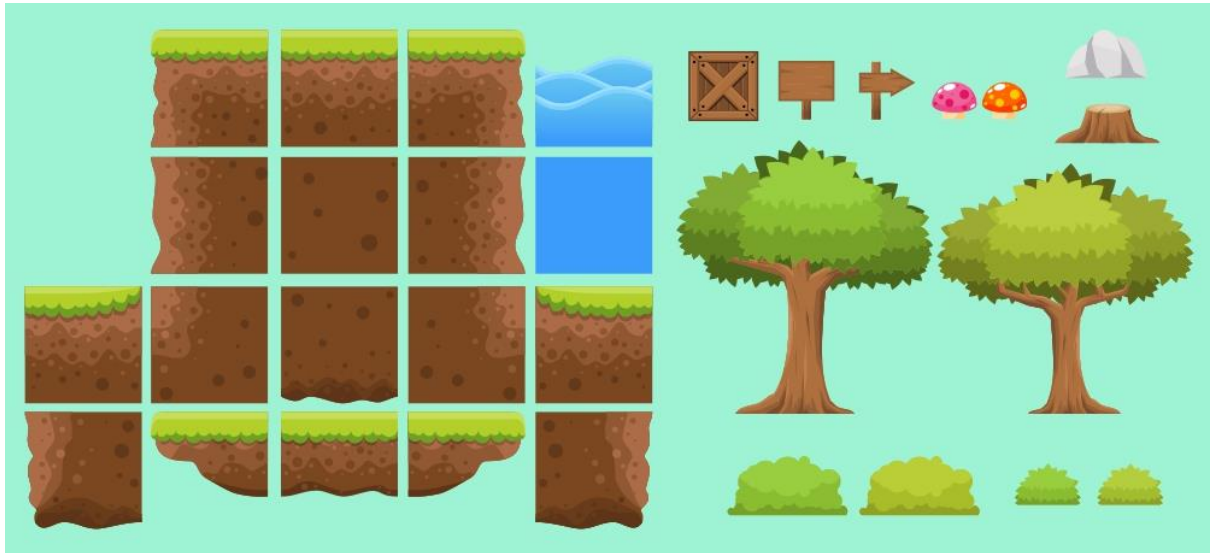
Figura 29 - A *Sprite-sheet* do protagonista do jogo *Megaman* (1987), Capcom.²¹



²¹ Disponível em: <https://jessebull.wordpress.com/tag/pixel-art/>

Tilesets, diferente de *sprites*, são conjuntos de peças, chamadas de “*tiles*” (ladrilhos em inglês), que não servirão para animação, e sim para a montagem do cenário no qual o protagonista irá interagir. Também são feitos a partir de uma grade uniforme, normalmente com células quadradas, embora há *softwares* que permitam não utilizar grades.

Figura 30 - Uma Tileset básica de um cenário²²



Os elementos sonoros também são considerados assets, porém são mais simples de se explicar e categorizar. Em jogos normalmente há entre dois e quatro tipos de som. Existe a *música de fundo*, essencial para esse projeto, mas também há os *sons de ambiente*, que simulam pássaros, água corrente e outros elementos de fundo, com o objetivo de aumentar a imersão do jogador no cenário. Há os *efeitos sonoros*, que tocam em ocasiões particulares, como o jogador pulando, ou um vaso sendo quebrado por ele, e por último, há a voz dos personagens, para jogos onde existe diálogo, o que não é o caso do projeto.

Jogos mais simples utilizam apenas a música e os efeitos sonoros, enquanto jogos complexos têm os quatro tipos de som. Para aumentar a imersão do jogador no projeto, foi decidido utilizar três. A música, essencial pela definição do jogo, como já havia sido dito, os efeitos sonoros e os sons de ambiente, para trazer um elemento natural e realista ao jogo e aumentar a imersão no cenário.

²² <https://opengameart.org/content/platformersidescroller-tiles-and-backgrounds>

4.3.2. Programação, Implementação, Erros e Testes

Videogames, como mídias interativas e digitais, são em sua essência softwares, programas de computador. Mesmo se utilizar ferramentas que eliminam a necessidade de conhecer e escrever linguagens de programação, ainda há um exercício de lógica e programação ao montar as condições e ações que formarão o produto final.

Assim como quaisquer outros programas de computador, estão sujeitos a erros de programação que fazem com que o jogo não funcione da maneira determinada, causando desde problemas sérios como o jogo simplesmente parar de funcionar até problemas que alteram suas mecânicas, como por exemplo fazer com que o jogador pule alto demais.

Também são possíveis erros no design do jogo, como fases confusas e impossíveis de se concluir devido ao mau planejamento por parte dos desenvolvedores. A criação de um jogo com desafios injustamente difíceis ou insatisfatoriamente fáceis também é possível.

O melhor método para se resolver os dois problemas apontados é a execução de vários testes ao decorrer da criação do jogo. Tais testes são realizados da seguinte forma: jogando o jogo várias vezes, tentando todas as possibilidades, encontrando os erros que devem ser resolvidos, resolvendo-os e testando de novo.

Esta possivelmente é a parte mais importante da produção do jogo, e é essencial para que o produto final seja o melhor possível.

5. DESENVOLVIMENTO

No capítulo atual, os elementos criados para o projeto serão discutidos um a um, explicando a lógica por trás de cada decisão tomada em seu desenvolvimento.

5.1. Pré-produção

Os elementos que moldaram o produto final, incluindo a história e Concept-art dos personagens e outros objetos presentes no jogo.

5.1.1. História

A história do jogo, para ser desenvolvida, partiu de uma ideia geral sobre o tema e a quantidade de personagens que estariam envolvidos no jogo. Foram definidos como seriam os protagonistas, os inimigos, e levado em consideração o tema musical do jogo, e então foi criada uma trama que os conecte de maneira satisfatória.

O protagonista, um menino de cabelos ruivos e de personalidade animada, foi chamado de *Boogie Jr.*, com base no estilo musical *Boogie-Woogie*, popular no início do século XX, e na música *Boogie Wonderland*, de 1979, pelo grupo *Earth, Wind & Fire*. Também foi criada uma irmã para o protagonista, com traços afrodescendentes e uma personalidade mais tímida, e chamada de *Woogie*, inspirada pelo mesmo gênero musical, porém, devido ao limite de tempo, seria difícil de implementá-la no jogo.

Há um exército de robôs que lembram caixas de som e alto-falantes distorcidos e tortos, que representam a cacofonia e o caos, e são os oponentes que o jogador deve enfrentar. Eles se opõem ao mundo onde os personagens habitam, formado por pessoas ligadas à música e à harmonia, e com temas musicais pontuando o cenário.

Conectando todas as ideias determinadas, e levando em consideração o limite de tempo, que reduz o escopo do projeto à apenas uma fase e possivelmente impede a exibição de personagens dentro do jogo em si, foi criada uma breve trama para justificar as ações do protagonista, *Boogie jr.* Ela é:

Woogie, a irmã de *Boogie jr.*, desapareceu misteriosamente da noite para o dia, ao mesmo tempo que misteriosos robôs começaram a invadir a pacífica ilha onde sua família morava. Em busca dela, *Boogie jr.* deve derrotar os robôs cacofônicos que ameaçam a paz da região.

5.1.1.1. O nome

Na hora de se escolher o nome do projeto, originalmente se teve a intenção de escolher um nome que fizesse referência aos personagens do jogo, e aos jogos desenvolvidos pelo estúdio *Rareware* e seu sucessor espiritual, *Playtonic* (*Banjo-Kazooie*, *Yooka-Laylee*), nessa etapa inicial, se pensou chamar o jogo de *Boogie-Woogie*. Porém, eventualmente foi preferido escolher um nome mais curto e com menos relação aos jogos mencionados, para que ele possa ser tratado como uma entidade própria, em vez de uma referência. Outro motivo foi a dúvida de se haveria tempo para implementar a personagem *Woogie*.

O nome do projeto foi definido como *Encore*. É um termo muito utilizado na música e no *show business*, sendo um evento onde os artistas retornam ao palco para dar uma performance adicional, ao final da apresentação. É uma palavra em francês que significa “outra vez, de novo”, e é utilizada em países de língua inglesa da mesma forma que no Brasil se utiliza o termo “Bis”.

Foi escolhido por conta de seu contexto musical e por conta de seu significado. Ele representa não só o tema do jogo, como sua natureza como um projeto de diplomação, uma chamada para o “artista” realizar uma última “performance”, ou seja, para o Designer realizar um último projeto como estudante. Não é *Encore* por ser apenas musical, mas por ser um ato de se projetar mais uma vez, de apresentar mais uma vez.

Figura 31 - A marca do jogo



A marca do jogo foi feita com o objetivo de capturar o estilo visual do jogo em sua totalidade. A quebra de padrões de tipografia e de grades e medidas em favor de algo feito manualmente e propositalmente distorcido transmite a ideia de que o jogo é descontraído e voltado para o entretenimento, e faz referência a outros jogos do gênero plataforma, igualmente descontraídos.

A pintura com uso de sombreamento e luz reflete o estilo visual do próprio cenário de dentro do jogo e o aspecto manual do projeto inteiro, fazendo com que cada letra pareça ser feita de uma das pedras que podem ser vistas no cenário do jogo. A letra “O” foi substituída por um símbolo circular azul e disforme com uma versão estilizada de uma colcheia em amarelo inserida no meio, sendo esse símbolo algo que será importante para o *gameplay* e parte da representação visual do personagem principal.

5.1.2. Personagens

Os personagens principais do jogo, Boogie e sua irmã Woogie, estão diretamente atrelados a todos os aspectos do projeto. O desenvolvimento da sua aparência foi o maior guia para o desenvolvimento visual do projeto inteiro, e a existência deles reflete a história. Ao discorrer o processo de criação dos dois, é possível mostrar um panorama do desenvolvimento visual do projeto inteiro.

5.1.2.1. Boogie

O protagonista começou a ser desenvolvido junto com o *High Concept* do jogo. Por ser o personagem com quem o jogador terá mais contato ao decorrer do jogo, foi necessário criar um design agradável, único e que capturasse a essência do personagem, e que trouxesse clareza e contraste com o fundo, para facilitar o gameplay.

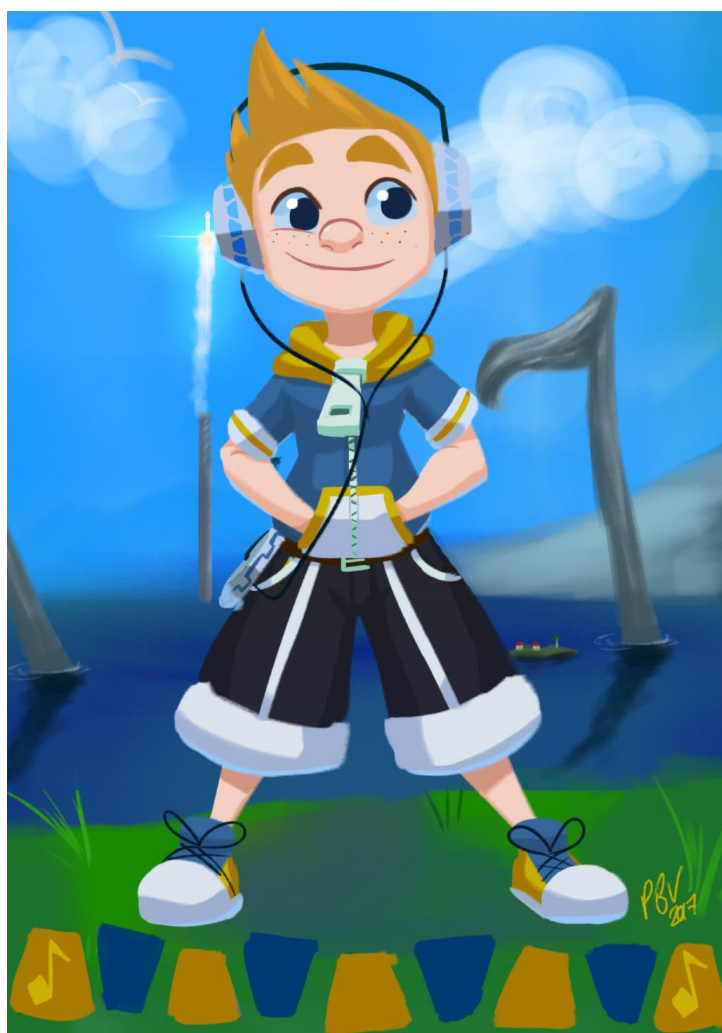
Figura 32 - Versões 1 e 2 do protagonista. A versão 2 foi utilizada para o High Concept



O personagem começou a ser desenvolvido inspirado em padrões visuais comumente associados a festas de música eletrônica, com roupas neon e óculos escuros estilo persiana. Desde o começo havia a intenção de dar a ele um topete espetado e grandes fones de ouvido, além de uma bermuda larga e tênis sem meia.

Pouco depois do desenho inicial, foi criada uma versão almejando um aspecto visual mais infantil e amigável. Foram removidos os óculos escuros e aumentados os olhos, dando mais foco nas expressões faciais. O rosto foi arredondado e a cabeça tornou-se maior em relação ao corpo, com o objetivo de dar ainda mais a sensação de um personagem de desenho animado. Além disso, criou-se uma jaqueta de mangas curtas, com um capuz e um zíper extremamente grande.

Figura 33 - O High-concept do jogo, outra vez.



Esta versão foi utilizada no High Concept, utilizando principalmente as cores Azul, Amarelo, Preto e Branco. Eventualmente, o cabelo do personagem seria alterado para ruivo. A bermuda foi feita em preto com faixas brancas frontais. A cor escura da bermuda permite que o personagem se destaque do fundo até em cenas bastante claras, enquanto as faixas brancas servem como um guia visual para o jogador. Durante o jogo, o protagonista quase sempre está virado de lado, correndo, pulando entre outras ações.

A cor escura da bermuda faz com que fique difícil de perceber qual perna do protagonista está mais “à frente” na tela, o que causa ilusões de ótica indesejadas nas animações. As listras brancas permitem que o jogador compreenda qual perna é qual, tornando as animações mais claras e eliminando as ilusões. As animações serão discutidas em mais detalhes durante a descrição dos assets.

Figura 34 - Teste de estilo para o protagonista.



Conforme a ideia visual do jogo foi progredindo e surgiu a ideia de criar cenários quadrados que fizessem referência ao aspecto de “tiles” quadrados dos jogos em 2D, o design do personagem retornou à uma ideia da versão original, de se fazer uma cabeça de forma quadrada, e bordas evidentes. Porém, se manteve o rosto expressivo e amigável.

Após determinado tempo de planejamento, na hora de executar a pesquisa de software, foi realizado um teste de pintura com o programa *Krita*, para verificar a diferença de tempo que seria gasto para pintar nele ou no *Photoshop CS6*. Com esse teste de pintura, foi tomada a decisão de se utilizar uma pintura detalhada, com detalhamento de volume suavizado.

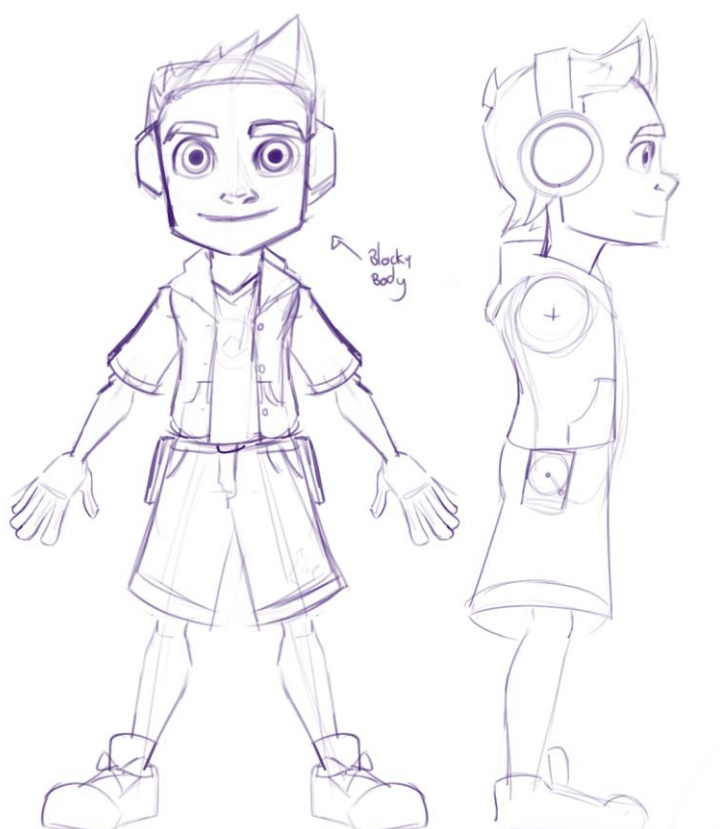
Figura 35 - Versão 3, parte do teste feito no Krita. A imagem inteira será discutida mais adiante.



Durante o teste, foi desenvolvida a terceira versão do personagem, mais próxima da versão final. A jaqueta de manga curta foi substituída por uma camiseta branca com detalhes em vermelho, e um colete azul com detalhes em branco e amarelo. É introduzido o ícone que foi utilizado na marca do jogo. A barra da bermuda foi alterada para amarelo em vez de branco, e as cores dos tênis do personagem inverteram de posição.

O design do personagem começou a se focar na facilidade para se animar e reproduzir no jogo. Para isso, as roupas passaram a valorizar a diversidade de cores mais do que a quantidade de detalhes pequenos, foi retirado o Zíper. Também foi retirado o tocador de CDs que ficava acoplado ao lado da bermuda, pois como os *assets* do personagem deveriam ser espelhados no jogo, era recomendado um design mais simétrico possível.

Figura 36 Teste final do personagem.



Um último desenho foi feito, mais detalhado que o desejado para o produto final, porém com o objetivo de definir as proporções finais do personagem. A relação entre o tamanho da cabeça e do corpo, o rosto do personagem em perfil, além do formato final da cabeça dele, encontrando um meio-termo entre o formato natural de um rosto e a ideia original, mais angular.

5.1.2.2. *Woogie*

A irmã e melhor amiga de Boogie Jr. exerceria um papel vital na história, sendo uma personagem que o jogador poderia controlar, oferecendo variações na história e no modo de se jogar o jogo pois cada um haveria habilidades diferentes. Infelizmente, devido a limites de tempo, não seria possível criar os assets necessários para colocá-la no jogo, e então ela tornou-se um personagem de fundo para o projeto de diplomação.

Figura 37 - Primeiro teste para o rosto de Woogie.



Desde o início, a intenção era utilizar elementos gráficos para definir a conexão com o protagonista. Woogie deveria ter um design semelhante o suficiente ao de Boogie Jr., referenciando o fato de que são irmãos, próximos e que cresceram juntos em um mesmo ambiente. Ao mesmo tempo seu estilo visual deveria contrastar com o dele, para demonstrar suas personalidades e habilidades diferentes.

Para alcançar o efeito desejado, o design de Woogie varia elementos do design de Boogie, desde cores até detalhes como o comprimento de peças de roupa e como a personagem as utiliza. Enquanto as roupas de Boogie Jr. são infantis, com uma bermuda larga e um colete colorido, as roupas de Woogie são mais reservadas e escuras.

A bermuda se torna uma calça que alcança os tênis, a camiseta branca se torna uma camisa escura e o colete azul e amarelo se torna um casaco branco e vermelho, amarrado na cintura. Detalhes menores foram mudados, como o fone de ouvido no pescoço e não nas orelhas, e faixas na lateral da calça em vez de na parte frontal. Todas as mudanças foram feitas com o objetivo de torná-la diferente de Boogie Jr., mas ainda semelhante o suficiente para que o vínculo familiar dos dois seja plausível.

Figura 38 - Versão final de Woogie



Embora a personagem talvez não apareça no projeto do jogo para a diplomação, criar uma peça visual que represente a versão final dela foi importante por motivos de arquivamento, e também para ser capaz de capturar o estilo de desenho que será utilizado no próprio jogo. A peça foi feita após ter sido definido qual software utilizar, o *Photoshop CS6*, e nela foram definidas as técnicas de pintura e desenho que seriam utilizadas para a produção dos *assets* finais do videogame.

5.1.3. Cenário

O aspecto visual do cenário é vital para ser capaz de representar o estilo desejado para o jogo, mais até que o design dos personagens. É no cenário que serão transmitidas as intenções para o jogo, desde o aspecto de dia claro e ensolarado até a formação em “blocos” retangulares que faz referência a jogos antigos e à natureza do cenário como algo modular constituído de imagens quadradas (“azulejos”, “tiles”), algo que foi definido conforme o cenário foi desenvolvido.

Figura 39 - Primeiro esboço de cenário



O primeiro desenho conceitual do cenário foi feito não com a intenção de definir o cenário em si mas como um pequeno teste de tamanho, comparando o tamanho do personagem em relação ao resto da cena. Porém, a partir desse desenho surgiram ideias que dirigiram a produção final. Nesse esboço, definiu-se um uso considerável de cachoeiras e surgiu a ideia de um cenário baseado em formas retangulares.

Figura 40 - Teste realizado no Krita, em sua forma completa.



A partir do teste no Krita, uma visão mais definida da direção a ser tomada para o cenário foi formada. Cores vivas, incluindo gramados bem verdes, um mar extremamente azul e um céu claro, para transmitir a sensação de um dia bastante ensolarado. Além disso, a montagem do cenário em blocos individuais, retangulares e de tamanhos diferentes, e a implementação de espirais em seus desenhos para dar um aspecto quase artificial ao cenário.

O cenário também é pontuado de referências musicais, como as estruturas gigantescas ao fundo imitando a parte superior de uma colcheia, símbolo utilizado em partituras musicais e associado à música. Essa ideia já havia sido testada no *High-Concept*, porém realizadas com mais eficácia na peça acima.

Figura 41 - Símbolo de uma Colcheia.



Outra ideia testada no *high-concept* e concretizada durante a realização da ilustração é a de casas e prédios de estuque, com portas e janelas de madeira e telhas vermelhas. A intenção é criar um aspecto “rural” ao cenário, dar a sensação de que o protagonista mora longe da cidade grande, mas que ainda haja civilização o suficiente no ambiente onde ele se encontra para que seja plausível que ele se vista de maneira moderna, tendo acesso a coisas como fones de ouvido, mesmo estando em um cenário praticamente selvagem.

5.1.4. *Miscelânea*

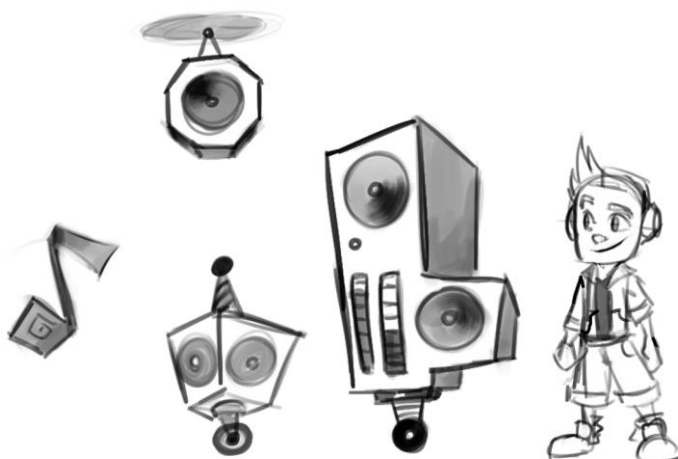
Alguns elementos tiveram seu desenvolvimento visual menos elaborado, então há menos o que ser dito sobre eles, porém ainda há informação o suficiente para que seja importante mencioná-los.

5.1.4.1. *Inimigos*

Parte do desafio em um jogo de plataforma são os vários inimigos que o jogador encontra. Em termos de funcionalidade, os mais simples em um jogo como Super Mario não passam de obstáculos que se movem. Encostar neles mata o personagem, a não ser que ele pule em cima deles, eliminando-os sem sofrer danos. Há variações, no entanto, com inimigos mais complexos requerendo estratégias diferentes para serem derrotados.

Para o projeto, foram criados três tipos de criaturas, para aumentar a variedade de obstáculos que o jogador irá enfrentar. Os três seguem a história, sendo parte de um exército de robôs que causam cacofonia (sons desagradáveis, sem harmonia) e caos.

Figura 42 - Esboço dos inimigos, ao lado do personagem para comparação de tamanho.



Os vínculos ao som e ao caos foram os elementos mais importantes a se considerar em seu design. Como sua única função em termos de história é causar cacofonia, os robôs são apenas grandes alto-falantes capazes de se mover sozinhos por meio de rodas ou hélices, sem nenhum elemento visual que os aproxime de um ser humano ou animal. O aspecto caótico da cacofonia também é representado visualmente, pois os robôs são disformes, tortos, construídos de modo que pareçam não ter lógica.

5.1.4.2. *Boogie Adolescente*

Durante a fase inicial de planejamento, surgiu a ideia de fazer os personagens mais velhos, como adolescentes ou jovens adultos, em vez de crianças. Porém, devido ao estilo do jogo, fazer personagens mais velhos retiraria o aspecto desejado e possivelmente não combinaria com a experiência desejada.

Um personagem de estatura mais alta está melhor colocado em algum jogo com foco maior em ação, acrobacias, e uma história mais madura. Para um jogo simples e infantil, um personagem de baixa estatura, expressivo e cartunesco, combina mais (por exemplo, o personagem Mario, da Nintendo).

Ainda assim, um pequeno esboço de como o personagem *Boogie* seria quando mais velho foi realizado, como registro da ideia, podendo ser futuramente aplicado em uma ocasião em que o projeto de jogo possa ser retomado.

Figura 43 - Boogie adolescente



5.2. Produção

Neste tópico, são percorridas as várias partes que compõem o produto final, desde o desenvolvimento das peças visuais dele até elementos de sua mecânica que tornam possível alcançar a experiência desejada.

5.2.1. Assets

As peças gráficas e sonoras que compõem o jogo tiveram uma lógica por trás de sua construção, nesse sub-tópico, os *assets* têm o seu desenvolvimento detalhado.

5.2.1.1. *Boogie*

A primeira etapa para a criação dos *assets* do protagonista é definir quais ações ele irá executar no jogo. Pular, cair, correr... cada tipo de ação que Boogie executa deve ser representada visualmente, por meio de animações ou imagens paradas. Portanto, definir as ações que o protagonista pode realizar no jogo é a primeira etapa para definir a quantidade de desenhos necessários. Para o escopo do projeto de diplomação, o personagem é capaz de: Pular, cair, correr, ficar parado e levar dano.

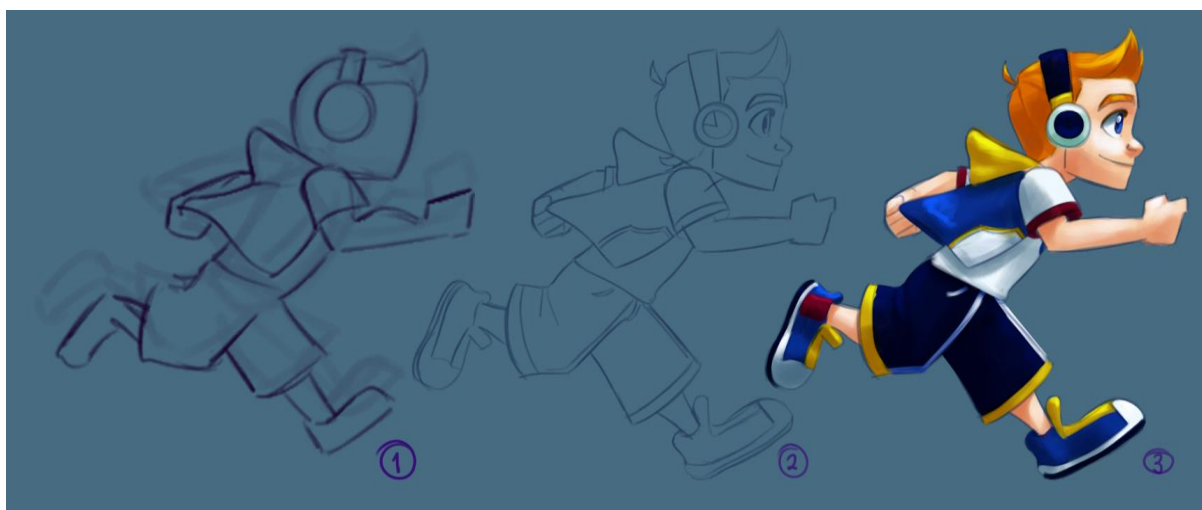
O desenvolvimento se iniciou com a animação para o personagem correndo. Ela deveria ser uma animação que pudesse ser repetida infinitamente de maneira imperceptível, formando um loop contínuo. Para isso, o último quadro da animação deveria progredir logicamente para o primeiro.

A técnica de animação a ser usada é a tradicional. Como ela implica em criar um desenho novo para cada quadro da animação, foi recomendado simplificar o nível de detalhamento e sombreamento do personagem, para facilitar a carga de trabalho. Porém, devido ao estilo visual desejado, foi decidido manter o sombreamento que pode ser visto na *concept-art*.

Para ser capaz de manter a complexidade visual, decidiu-se reduzir o máximo possível a quantidade de quadros, utilizando o mínimo necessário para dar a ilusão de movimento, por mais travada que a sensação seja. Em algumas ações, foi preferido não utilizar animações, utilizando apenas um único quadro.

A animação desenvolvida representa o protagonista correndo. Ela contém 6 quadros, que foram considerados o mínimo necessário para dar a ilusão de movimento. Ela foi desenvolvida da seguinte maneira, ilustrada pela imagem a seguir.

Figura 44 - Os 3 passos para a criação da animação.



1. Todos os frames da animação são esboçados de maneira rápida e sem muita preocupação com a forma final ou com detalhes. Desse modo é possível ter uma noção do fluxo da animação em pouco tempo, e fazer ajustes no movimento, na velocidade e na quantidade de quadros antes de partir para a versão definitiva do desenho.
2. Após a definição dos quadros, é feito o desenho mais detalhado de cada um. Nessa parte, é essencial manter consistência na proporção e nas formas do personagem, criando uma animação que progride suavemente.
3. Por último, os quadros são pintados utilizando as mesmas cores, tentando criar uma iluminação uniforme e quase realista, mantendo unidade entre os quadros não só em proporção e formas, mas no volume também.

No total, seis quadros foram desenvolvidos para a animação do personagem correndo. Os mesmos quadros foram usados de referência para as outras ações. Porém, devido ao prazo limite, elas não possuem animações associadas a elas, sendo apenas imagens individuais.

Figura 45 - Cada ação que o personagem pode executar. Da esquerda para a direita: Correr, pular, cair, ficar parado e sofrer danos.



A ação de pular consiste em duas imagens. Uma onde o personagem aparenta estar subindo, e uma onde ele aparenta estar descendo. A separação entre as duas “sub-ações” permite que a imagem do personagem descendo possa ser utilizada quando ele está caindo, independente dele ter pulado ou não.

Há a imagem para quando ele “morre” ao tocar em um obstáculo, representada como se o personagem estivesse levando um choque ou um susto. E por fim, uma imagem dele simplesmente parado, esperando a próxima ação.

Após completos, os desenhos precisavam ser preparados para o uso no projeto. Para isso, eles foram redimensionados para melhor encaixar na grade do jogo (que será explicada futuramente) e em seu cenário. Após o novo tamanho, eles também precisavam ser organizados em uma *spritesheet*, onde todas as células possuem o mesmo tamanho e acomodam qualquer um dos desenhos.

As imagens em si são salvas separadamente, incluindo cada quadro da animação, devido à maneira que o *GameMaker Studio*, utilizado para desenvolver o jogo, trabalha com imagens e animações. Porém, é importante que cada imagem separada possua o mesmo tamanho que as demais, para facilitar o alinhamento das animações e outras imagens no programa. Por isso a criação da *spritesheet*.

Figuras 46 e 47 - Spritesheet do protagonista, com a representação de cada célula abaixo, em tons de roxo.



5.2.1.2. Cenário

O *GameMaker Studio* é capaz de criar cenários inteiros com o uso de objetos, que são assets individuais com suas próprias mecânicas. O próprio protagonista é tratado como um objeto, assim como os inimigos que ele irá enfrentar, e qualquer outra coisa que necessite de um funcionamento especial.

Porém, para poupar espaço e melhorar a performance do jogo o máximo possível, foi preferível utilizar o método de *tiles* para a maioria das estruturas na fase, o que implicou em fazer com que todo o cenário seja capaz de se encaixar em uma *grid* específica.

O *tileset* para o cenário do jogo foi montado em uma grade de 128x128 pixels. As primeiras peças a serem feitas foram as peças do chão, as pedras. Foram feitos quatro tipos de pedra, cada um com três variações de tamanho. Há uma pedra básica, e uma versão escura para ser utilizada em cavernas. Há uma com espiral, para trazer o estilo das concept arts para o cenário, e uma com um alto-falante embutido, para trazer referência ao tema musical do jogo.

Figura 48 - Peças do cenário



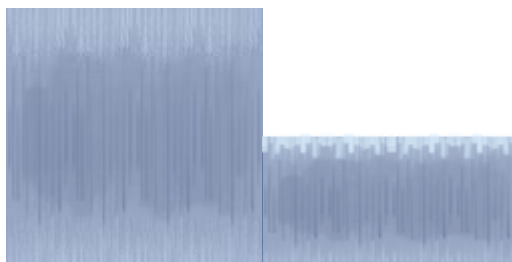
As peças com alto-falantes são animações que pulsam ao ritmo da música. Por serem animações, elas são tratadas como objetos, e salvas separadamente, uma vez que o programa não é capaz de trabalhar com *tilesets* animadas. Peças decorativas também são salvas como objetos, como por exemplo peças do gramado que serão colocadas por cima dos blocos de pedra para demonstrar quais partes do cenário o personagem pode andar em cima e quais ele atravessará.

As duas colunas, branca e preta, são salvas em separado como objetos pois serão plataformas que se movem no jogo, às vezes bloqueando o caminho, às vezes sendo plataformas onde o personagem deve mirar. Elas fazem referência às teclas de um piano.

As peças retangulares possuem bordas arredondadas, causando a sensação de que cada uma é uma peça individual por si só e impedindo que as formas mesquem entre si. Criar um cenário puramente retangular traria um aspecto artificial para ele, e vai contra o estilo visual mais “natural” do jogo.

Além dos auto-falantes, também foram feitas peças animadas para a criação de cachoeiras. São duas peças separadas, uma para a parte superior da cachoeira, e outra que compõe o resto de maneira modular. A divisão e a natureza modular permite a criação de cachoeiras de diversos tamanhos sem que seja necessário redimensionar a imagem, evitando distorções.

Figuras 49 e 50 - As peças da cachoeira



Após o desenvolvimento da maioria das *tiles* do cenário, foi realizado um teste de montagem, no próprio *Photoshop*, com um fundo provisório. A intenção do teste foi verificar a relação entre as peças em um cenário, a animação das cachoeiras, e medir o tamanho do personagem principal em relação à *grid*.

Foi definido que o protagonista teria aproximadamente, quando parado, a altura de três células da grade, ou seja, por volta de 384 pixels de altura. Após isso foi capaz de redimensionar todas as imagens pertencentes a ele, e montar a *sprite-sheet* vista no módulo anterior.

Ao realizar o teste, também foi visto que embora a intenção em criar peças com bordas arredondadas tenha cumprido o seu papel, ela trouxe um problema para a uniformidade visual do jogo. Pequenos buracos se formaram no cenário, trazendo um aspecto indesejado ao jogo. Para solucionar o problema, foi decidido criar, ao montar a fase, uma camada atrás das outras, utilizando as rochas escuras para tampar as frestas.

Figura 51 - Teste realizado no Photoshop, dimensões de 1920 x 1080 pixels.



Outro problema que aparentou surgir ao realizar o teste foi a questão do tamanho. A grid de 128x128 pixels implicou em imagens muito grandes para a resolução alvo do jogo, 1920x1080. O cenário estava muito “próximo” do jogador, e não era possível ver muito além do personagem. Para a jogabilidade desejada, incluindo a velocidade do personagem e altura de seu pulo, é melhor que o jogador possa enxergar exatamente o que há à frente do personagem, os obstáculos e plataformas, para que ele não jogue às cegas. Porém, o próprio Game Maker possui ferramentas para solucioná-lo.

Um último problema que foi percebido foi que infelizmente as peças do cenário tornavam ele confuso, então algumas foram re-criadas para o produto da apresentação.

5.2.1.3. *Sons e Música*

Para o escopo da diplomação, foi escolhido utilizar músicas disponíveis na *internet* para compor o jogo, pois a criação de músicas novas para o projeto demanda recursos indisponíveis no prazo necessário. As músicas foram escolhidas o melhor possível para demonstrar o estilo musical que o jogo deveria ter. Um total de três músicas foi selecionado, duas para a fase e uma para o menu do jogo.

Uma das músicas que tocam durante a fase é a música *Happy Boy*, do artista Harrison, que já foi mencionada nas referências musicais. Originalmente ela era apenas referência, mas devido à demanda por uma música já existente, ela foi escolhida como a mais próxima do estilo musical desejado para o jogo, e a que combina mais com o tema da fase e a sensação desejada.

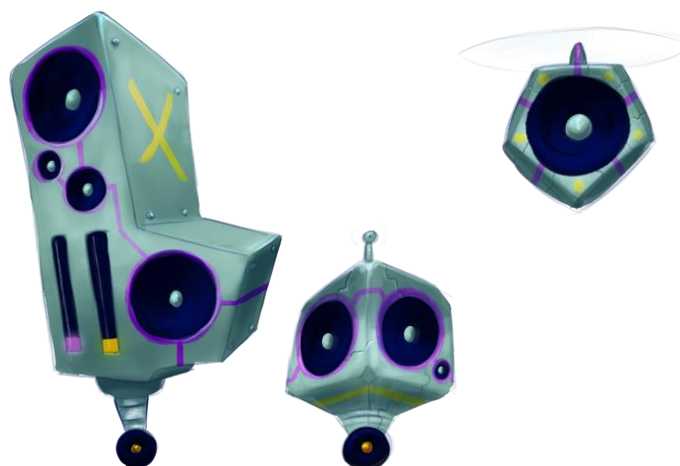
A segunda música é apenas um remix da anterior, pelo artista Shad Icre. É uma versão mais “caótica” da música original, com a repetição de notas e que traz uma sensação mais tensa. A intenção é que essa versão substitua a original quando o jogador se aproximar de algum robô inimigo, se apresentando como uma versão “errada” e “cacofônica” da música, como se os robôs estivessem manipulando a harmonia do cenário. Assim, os inimigos se apresentam como um problema para o mundo onde eles se encontram, como se eles realmente tivessem um papel na história além de serem apenas obstáculos para o jogador.

O menu principal do jogo demanda uma música menos intensa, que sirva a função de introduzir o jogo ao jogador e gerar uma expectativa, empolgando o jogador. Para isso, a música escolhida deveria trazer a sensação de que está sempre progredindo, como se estivesse lentamente avançando até um “pico” musical, um clímax, que nunca chega. Esse “pico” musical seria então o próprio jogo após o jogador sair do menu. O objetivo é fazer parecer com que o jogo estivesse esperando o jogador começar. Para alcançar essa função, foi escolhida a música *Heartbeat*, do artista JEDDI, editada para remover a última parte da música, mantendo apenas a parte inicial.

As três músicas tiveram suas batidas por minuto medidas, valor importante para sincronizar o jogo a elas.

5.2.1.4. Inimigos

Figura 52 - Versões finais dos robôs



Os três tipos de robôs foram implementados no jogo para oferecer variedade nos obstáculos que o jogador enfrenta. Cada um deles possui sua própria maneira de interagir com o cenário e oferece um desafio diferente para o jogador.

Os detalhes em roxo e rosa foram escolhidos para fazer um contraste entre eles e o azul do protagonista, e suas formas em linhas diagonais e círculos os separa do mundo ao redor deles, trazendo a ideia de que eles não fazem parte do lugar onde eles se encontram. Porém, essa diferença visual não é grande o suficiente para que eles não pareçam parte do projeto.

5.2.2. Programação

Esse sub-tópico relata as decisões e processos relativos à programação do jogo, incluindo soluções encontradas para a implementação de mecânicas que permitam alcançar os objetivos para o jogo.

5.2.2.1. Fase de Testes

Uma das primeiras etapas para se montar o jogo no *GameMaker Studio* é criar uma fase onde é possível testar todos os elementos que comporão o jogo, que o jogador não poderá acessar no produto final. Nela, os objetos, mecânicas e sistemas são implementados um a um e o desenvolvedor é capaz de testar o funcionamento deles antes da montagem do jogo final.

A primeira coisa a ser feita na fase de testes é um pequeno espaço para testar as mecânicas do personagem. É onde se testa a altura do pulo, a velocidade na qual o protagonista corre e se as animações estão funcionando corretamente. Também é onde se testa o Zoom da câmera do jogo, que será descrito no tópico seguinte. A parte mais importante é a criação do sistema de colisões, que define onde o jogador pode atravessar e onde o jogador para de se mover (piso, teto entre outras coisas).

O segundo teste é a implementação de plataformas em movimento e o sistema de sincronização do cenário ao ritmo da música, verificando como o protagonista interage com tais plataformas e resolvendo possíveis problemas nessa interação. Também foram implementados os objetos decorativos, pois seus *assets* já estavam prontos no momento do teste.

O terceiro teste é a implementação de obstáculos e outras situações onde o protagonista pode “morrer”, assim como a própria consequência desse ato de “morrer”. São implementados os inimigos e objetos invisíveis que matam o protagonista quando ele os toca, para situações onde o jogador cai em buracos. Também é implementado o sistema de *checkpoints*, que são os locais onde o personagem irá retornar após “morrer”. Por último, é implementado o sistema de recompensas e vitória, incluindo itens que aumentam a pontuação do jogador e o ponto final da fase, onde o jogador vence.

Com todos os sistemas montados e testados na fase de testes, é possível fazer o jogo propriamente dito.

5.2.2.2. Câmera, Zoom, Teste de Tamanho

Como descrito no tópico do desenvolvimento do cenário, as imagens criadas para o jogo estariam, em seu tamanho original, muito grandes para permitir a experiência ideal para o jogador, que precisa ser capaz de enxergar as plataformas à frente do personagem, mas o *Gamemaker studio* é capaz de resolver o problema por meio de Zoom.

Testar o melhor tamanho para o cenário foi uma questão de observação. Alguns valores para o Zoom foram colocados e dependendo do resultado visual foram descartados, até encontrar a melhor proporção para o jogo.

O primeiro teste foi feito com o Zoom a 100% do tamanho original, comprovando que, para a experiência desejada, realmente é muito próximo. Quando o jogador pula, por exemplo, se torna impossível de ver o que há em baixo do personagem, o que é essencial para um jogo de plataforma.

Figura 53 - Teste 1 - 100% do tamanho



O segundo teste põe o jogo a um zoom de 50% do tamanho original, e de quebra já elimina os problemas de mira e antecipação, permitindo que o jogador seja capaz de pular sem correr o risco de cair por não enxergar onde está indo. Em muitos casos, esse tamanho já seria o ideal, porém foi preferível reduzir um pouco mais.

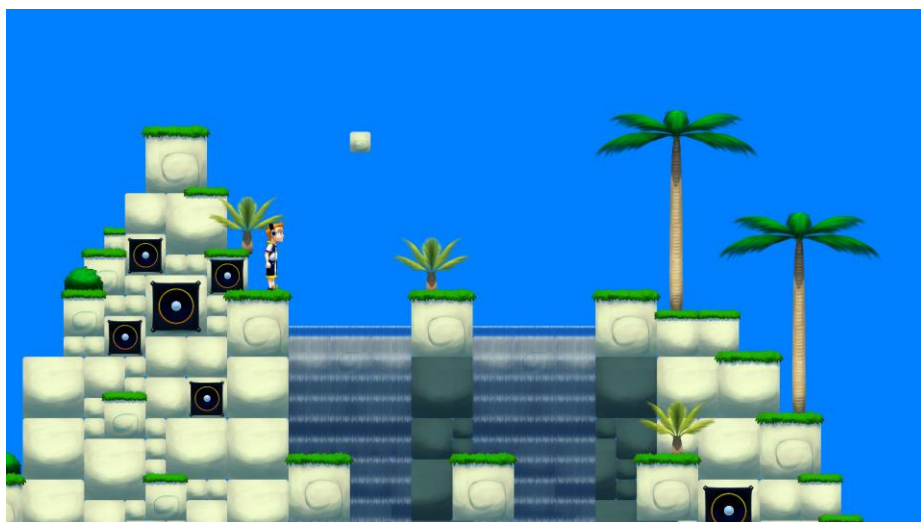
Figura 54 - Teste 2, à 50% do tamanho



O teste final foi realizado à um terço do tamanho original, e foi o melhor resultado para o tipo de experiência desejada para o jogador. A intenção é oferecer uma experiência onde o jogador não precise parar para pensar, e possa manter o fluxo de movimento e “se deixar levar” pelo ritmo da música.

Em muitos jogos de plataforma, o jogador geralmente alcança uma região onde há um obstáculo a ser enfrentado, ou um enigma a ser solucionado, para depois pensar em como solucioná-la. Isso implica em uma pausa no movimento do personagem, uma quebra no fluxo. Numa tentativa de resolver esse problema, permite-se que o jogador seja capaz de ver o obstáculo à uma distância maior, para que ele seja capaz de pensar na solução do problema antes de alcançá-lo.

Figura 55 - Teste 3, à 1/3 do tamanho



5.2.2.3. Ritmo

Implementar um sistema de ritmo ao jogo, para que o jogo esteja diretamente vinculado à música de fundo, foi uma tarefa essencial para que ele ainda se configure no gênero de plataforma e ritmo, mesmo sendo diferente da maioria dos jogos do gênero. Foi importante ser capaz de criar um sistema que consiga vincular a fase inteira à música de maneira automática, sem que seja necessário programar novos valores para cada peça dentro da fase toda vez que se mude a música.

Para isso, foram feitas três fórmulas matemáticas. Uma para calcular a duração de uma única batida rítmica da música, uma que utiliza esse valor para alterar a velocidade de animações, e uma que utiliza esse valor para sincronizar o movimento de plataformas e inimigos ao ritmo da música.

A primeira fórmula pega a quantidade de batidas por minuto da música que está tocando e divide por 60, para obter a quantidade de batimentos por segundo. Ela então divide esse segundo valor pela quantidade de quadros por segundo, que é a velocidade em que o jogo está rodando. Em situações ideais, essa velocidade é também de 60 quadros por segundo. Assim é capaz de se ter a duração de um único batimento em relação à velocidade do jogo, sendo esse o valor final da fórmula.

Figura 56 - Representação da primeira fórmula no código do jogo.

```
1  ///Tempo Index
2  if room = testmap1
3  { global.music_bpm = 114;
4  audio_play_sound (level1_bgm, 1 , true)
5  }
6
7  // Global Calculations
8
9  global.music_bps = global.music_bpm / 60;
10 global.music_sBeat = global.music_bps / 60;
11
12 // Beat Counter
13 global.beatCounter = 0;
```

A segunda fórmula é aplicada individualmente em cada objeto cuja animação deve ser sincronizada à música. Ela determina a velocidade da animação como o resultado da fórmula anterior multiplicado pela quantidade de *frames* há na animação. Assim se tornou possível sincronizar a animação dos alto-falantes, cachoeiras e inimigos ao ritmo da fase.

Figura 57 - Representação da segunda fórmula no código do jogo, especificamente no objeto do alto-falante maior, no cenário.

```
1 /// Rhythm Sync
2 sprite_index = boom_1
3 image_speed = global.music_sBeat * sprite_get_number(sprite_index)
```

A terceira fórmula, a mais difícil de fazer funcionar corretamente, cria um contador que vai até 4 antes de voltar ao menor valor, indicando as quatro batidas de um compasso musical, às quais serão vinculados os movimentos das plataformas, inimigos, entre outros. Por exemplo, uma plataforma pode se mover até certa posição na primeira batida de um compasso, e retorna à posição original na terceira.

Para isso, o contador começa em 0, e vai aumentando de pouco em pouco, em incrementações do resultado da primeira fórmula (“X”), e quando o valor se torna maior que 4, ele é reiniciado. Porém, em vez de retornar a 0, ele retorna ao valor “X”.

Figura 58 - O código do contador.

```
1 /// Counter
2
3 global.beatCounter += global.music_sBeat
4
5 if (global.beatCounter > 4)
6 { global.beatCounter = global.music_sBeat };
7
```

Originalmente, foi feito com que o contador voltasse a 0 no momento que ele alcançasse 4, porém como “X” não era um número inteiro, ele não se tornava exatamente igual a 4 em momento algum, e, portanto, não era reiniciado. Portanto se fez com que ele se tornasse igual a 0 assim que ele ultrapassasse 4.

Ao resolver essa parte, surgiu um outro problema, o contador estava causando uma quebra na sincronização, a distância entre a batida 4 e a batida 1 do próximo compasso parecia ser mais comprida que as outras três. A solução foi fazer com que o contador retornasse para um valor maior que 0, especificamente o resultado da primeira fórmula.

À partir dessa fórmula, os objetos são capazes de arredondar o valor do contador em qualquer momento e executar determinadas ações quando esse valor bate com o número desejado (1, 2, 3, 4 ou 0), permitindo que eles se sincronizem à batida da música.

Figura 59 - O valor do contador sendo aplicado em uma plataforma móvel. O contador controla a variável que determina a direção do movimento da plataforma.

```

1  /// Variables - Counter
2  //Vars
3  if (round(global.beatCounter) = 1)
4  {vmov = -1}
5
6  if (round(global.beatCounter) = 3)
7  {vmov = 1}

```

5.2.2.4. Morte e Pontos

A natureza de um jogo propõe que haja um desafio, um objetivo a ser cumprido e obstáculos que possam impedir o jogador de cumpri-lo. Em jogos tradicionais, boa parte dos desafios são os inimigos, que, caso o jogador perca o confronto com eles (a mecânica desse confronto dependendo de como o jogo em específico funciona), causam a “morte” do personagem, que reinicia a fase e reduz a quantidade total de “vidas” que ele tem. Quando esse total chega a 0, o jogo acaba e o jogador perdeu.

Para a experiência desejada, o formato tradicional de se tratar a “morte” e a derrota do personagem não era adequado. O ato de reiniciar a fase implica em retornar tudo à posição inicial, incluindo a música, o que quebra o fluxo contínuo desejado para o jogo. Uma possibilidade para criar o efeito de morte sem reiniciar a fase era de simplesmente “teletransportar” o personagem até a posição desejada e diminuir o contador sem reiniciar a fase, porém a mudança súbita de posição também causa uma quebra no fluxo desejado.

Decidiu-se criar uma animação para a morte do personagem, fazer com que o personagem volte ao último *checkpoint* de maneira mais “natural” que simplesmente transportar o personagem imediatamente. Assim, mesmo havendo uma pausa na ação do jogador, pois ele deve esperar o personagem chegar ao *checkpoint*, o fluxo do jogo em si continua sem interrupções.

Com a criação desse novo tipo de “morte”, surgiu a ideia de também mudar o sistema de penalidade e recompensa do jogo. Em vez de haver um contador de “vidas”, o jogo passa a ter um sistema de pontos que dão uma nota final ao jogador, e ela define se o jogador venceu a fase ao chegar ao fim. A pontuação do jogador é aumentada ou diminuída quando determinadas situações acontecem, como o personagem morrer, por exemplo.

Dentro da fase, a nota atual do jogador é indicada no fundo do cenário, como parte orgânica do mundo. Ela é representada pela animação do sol, que muda de “humor” conforme a performance do jogador. Assim, o jogador consegue ter uma noção de seu resultado.

Figura 60 - O sol representando a nota do jogador.



6. RESULTADO

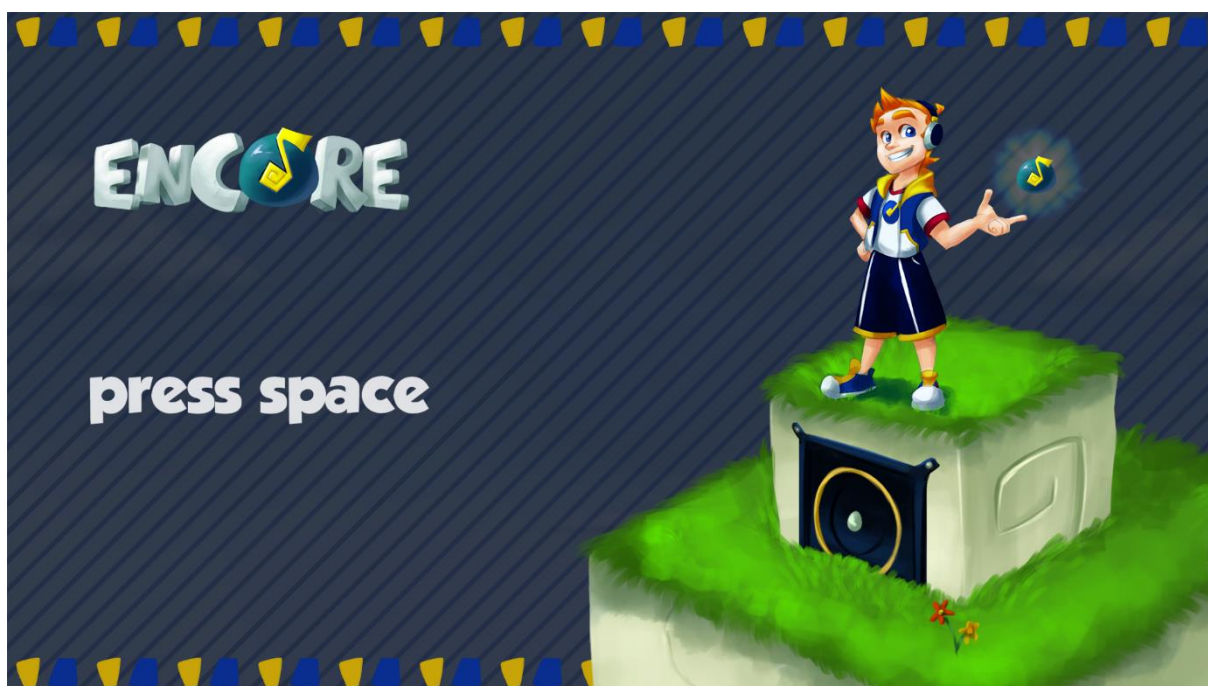
Todas as decisões tomadas e elementos criados durante os capítulos de conceituação e desenvolvimento foram aplicados em um protótipo do jogo, que inclui uma fase de demonstração para os seus sistemas. Nesse capítulo, o protótipo é descrito em três partes, a Tela Inicial do jogo, o Jogo em si e a tela onde o jogador recebe a sua nota, ao final da fase.

6.1. Tela Inicial

O menu inicial de um jogo tem a função de permitir que o jogador acesse o jogo em si, opções para o funcionamento do jogo e outras funções. Ele é a primeira coisa que o jogador vê ao entrar em um jogo, (embora haja exceções que colocam o jogador direto no jogo em si ao iniciar), e por isso é a primeira oportunidade de introduzir o jogador à experiência almejada, em termos de elementos gráficos e sonoros.

O foco para o menu principal foi introduzir a maneira que o jogo está fortemente atrelado ao ritmo da música que o jogador está ouvindo. Para isso, o menu é feito de várias animações que ocorrem de acordo com a batida do jogo. Ele também introduz o personagem principal e o tipo de item mais importante do jogo.

Figura 61 - Tela inicial



O menu principal também faz referência aos projetos anteriores realizados ao decorrer do curso, colocando seus personagens principais na ilustração. Eles também chegarão a fazer parte do jogo em si.

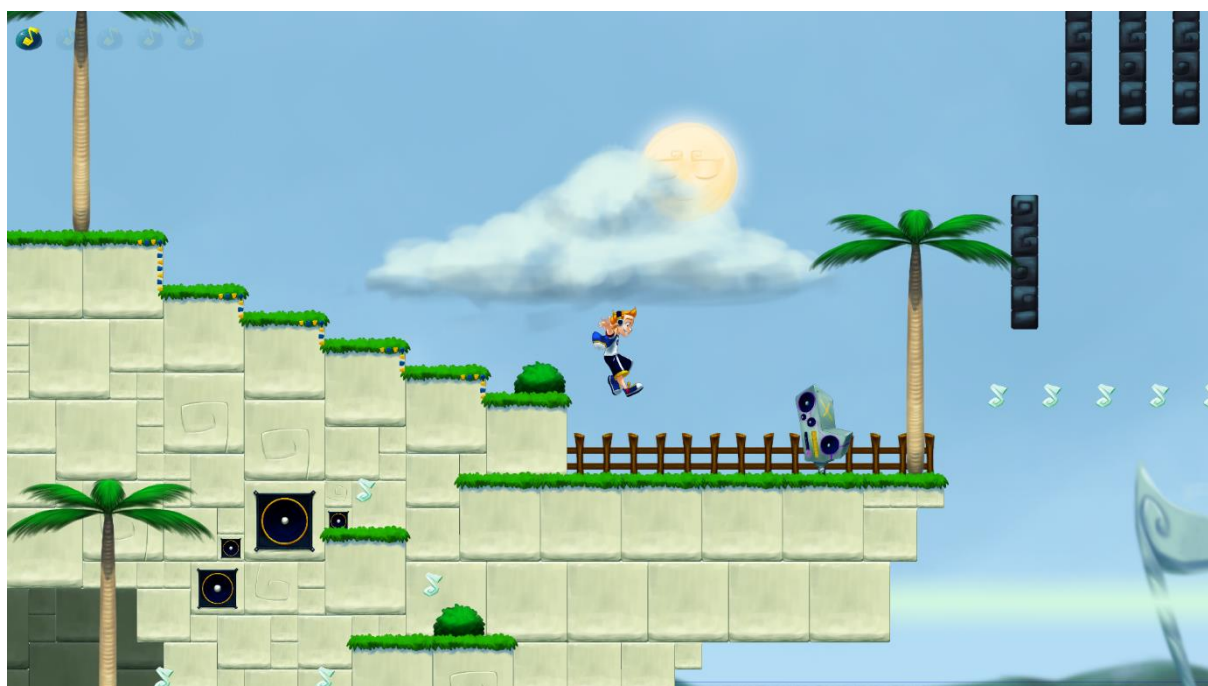
Figura 62 – Os personagens, juntos.



6.2. O Jogo

A fase jogável foi construída de modo que o caminho entre o começo e o fim seja simples, mas que tenha várias alternativas opcionais. O jogador é encorajado a procurar caminhos alternativos pois assim ele irá encontrar os itens necessários para poder completar a fase.

Figura 63 – Cena do jogo.

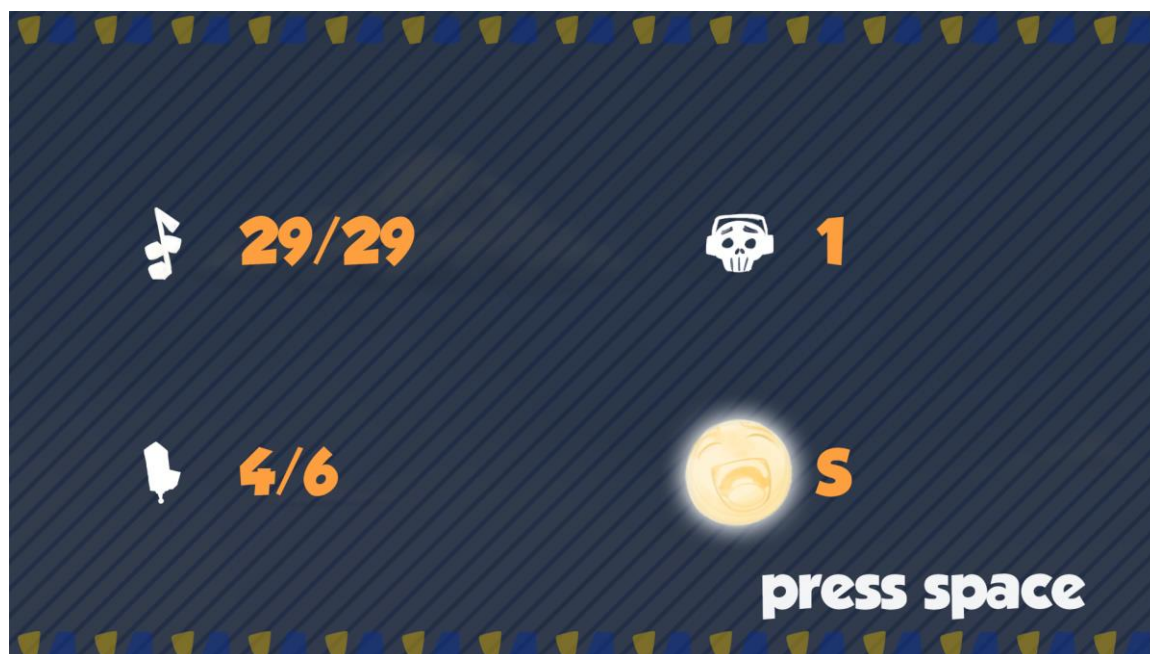


Ela foi criada de modo a introduzir as mecânicas do jogo de uma a uma, para então começar a misturá-las. O jogador começa experimentando a ação de pular, para então ser desafiado com buracos que podem mata-lo se ele errar o pulo. Então ele é introduzido ao tipo mais básico de inimigo e a plataformas móveis, e daí em diante as mecânicas começam a se misturar. O posicionamento das notas musicais foi pensado para incentivar a exploração, oferecendo pistas de que há coisas que o jogador ainda não viu.

6.3. Tela de Resultado

Após o jogador terminar a fase, é apresentado o resultado da performance dele. É mostrado a quantidade de notas musicais que o jogador coletou comparada com o total existente na fase, assim como os inimigos derrotados e a quantidade de vezes que o jogador “morreu”. No final, a performance do jogador é medida com um valor de 0 a 120, dando uma nota final que vai de “F”, pior nota, até “A”, “S” e por fim “SSS” se a performance tiver sido perfeita.

Figura 64 – A contagem de pontos ao final de cada fase



Vários fatores no jogo alteram a nota do jogador. Ao morrer, o jogador perde pontos, e enquanto ele estiver jogando, também perde conforme o tempo passa. Coisas como pegar notas musicais, matar inimigos e encontrar checkpoints aumenta a quantidade de pontos. O desafio e a consequência de “morrer” ou demorar demais a terminar a fase não está em ter que começar de novo ou ser “expulso” do jogo como na maioria dos jogos de Plataforma e de Ritmo, mas em sua performance ditando se ele falhou ou não.

7. CONCLUSÃO

Embora o projeto tenha sido desenvolvido dentro de várias limitações, incluindo tempo e recursos, ele foi capaz de cumprir seus objetivos. Ele se separa de outros jogos do gênero tanto visualmente quanto na própria mecânica, invertendo padrões de associação entre mecânica e música que eram comuns no gênero de plataforma e ritmo. É uma ideia que pode ser facilmente expandida com mais tempo, implementando novas maneiras de se associar o cenário à música, tornando-se um título capaz de introduzir algo novo ao seu meio.

Além disso, ele realiza homenagens a projetos realizados anteriormente durante o curso, desde seu estilo visual inspirado por outros projetos até referências diretas, e é um produto que culmina habilidades cultivadas pelos últimos 5 anos.

No final, o importante é que o método utilizado foi capaz de realizar um bom produto que atende o foco principal, a experiência desejada para o usuário. Limitações fizeram com que o jogo talvez não tenha tanta variedade quanto necessário, obrigaram a utilizar músicas não-autorais e tornaram necessários cortes e atalhos, mas ainda assim o resultado final é algo que pode ser experimentado de início ao fim, cumpre os seus objetivos e testou as habilidades demandadas.

8. REFERÊNCIAS

ESPOSITO, Nicolas. ***A Short and Simple Definition of What a Videogame Is.*** 2015.

Disponível em: <<http://www.utc.fr/~nesposit/publications/esposito2005definition.pdf>>

Acesso em: 26 nov. 2017.

HUNICKE, Robin; LEBLANC, Marc; ZUBEK, Robert. ***MDA: a formal approach to game design and game research.*** 2005. Disponível em:

<<https://www.cs.northwestern.edu/~hunicke/mda.pdf>>. Acesso em: 22 ago. 2017.

SIGNIFICADOS.COM. **Significado de jogo.** Disponível em:

<<https://www.significados.com.br/jogo/>>. Acesso em: 17 ago. 2017.

STORYTELLERS. **Já ouviu falar em high-concept?** Disponível em:

<<http://www.storytellers.com.br/2011/11/ja-ouviu-falar-em-high-concept.html>>.

Acesso em: 21 ago. 2017.

THE WRAP. ***What is a videogame? A short explainer.*** Disponível em:

<<http://www.thewrap.com/what-is-a-video-game-a-short-explainer/>>. Acesso em: 20

ago. 2017.